Available on CMS information server

CMS NOTE 2006/xxx

The Compact Muon Solenoid Experiment

CMS Note

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

30 November 2006

CMS Magnet Test Cosmic Challenge Phase-I Data Monitoring at the Fermilab Remote Operations Center

William Badgett, Jeffrey Berryhill, Ingo Bloch, Wade Fisher, Patrick Gartung, Kaori Maeshima, David Mason, Carsten Noeding, Alan Stone, Hans Wenzel, Yujun Wu, Francisco Yumiceva

Fermilab, Batavia, Illinois, USA

Zongru Wan

Kansas State University, Kansas, USA

Abstract

From the Remote Operations Center (ROC) at Fermilab, we participated in the quasi-online data monitoring for the first phase of the Magnet Test Cosmic Challenge (MTCC) conducted at CERN in August 2006. In this note, we describe our activities preceding, during and following the magnet test including our efforts to automate the data transfer, bookkeeping, processing and monitoring. We summarize the lessons learned, and provide recommendations for the subsequent phases.

1 Introduction

At the very end of July 2006, we were asked by the CPT MTCC coordinators at CERN to participate in quasi-online monitoring of MTCC Phase-I data from the Fermi National Accelerator Laboratory (FNAL) Remote Operations Center (ROC) which took place in the second half of August. Although we had been preparing for the various aspects of CMS online monitoring from the FNAL ROC in coordination with our colleagues at CERN for a long time, we needed to intensify our effort substantially in order to achieve the tasks in the short amount of time.

The following paragraph is an extract summarizing the MTCC-I experience from the document compiled by the CMS MTCC-I operation titled "Feedback from Magnet Test and Cosmic Challenge (MTCC) Phase I" [1]:

Ultimately the diligent work of hundreds of people over many years (aided by a little good fortune) transformed the "cosmic challenge" into a "cosmic success" for the CMS collaboration. Around 25 million "good" events were recorded with at least DT triggers and ECAL + TK in readout, of which 15M were at a stable field equal or greater than 3.8T. Data-taking efficiency reached over 90% for extended periods. Data transfer to some Tier 1 centers, online event display, quasi-online analysis on Meyrin site, and fast offline data-checking at Fermilab were some highlights of MTCC Phase I which offered a first hand taste of a CMS-like running experience.

Although we are 7,000 km away from the detector, we are very happy to be a part of the team working together. In the first section, we give a brief introduction of the CMS FNAL ROC and the strategy we employed for MTCC-I, followed by descriptions of "Data Transfer" (Section 2), "Automation and Bookkeeping" (Section 3), "Data Quality Monitoring" (Section 4), "Communications" (Section 5), and "Summary" (Section 6).

1.1 Introduction to the ROC at Fermilab

In the U.S. CMS [2] organization, the concept of a local remote operations center existed for a quite a while. The physical infrastructure of the FNAL ROC was built during 2005 in the northwest corner of the eleventh floor of Wilson Hall at Fermilab. The physical location of the FNAL ROC will relocate to the first floor of Wilson Hall during 2007 [3]. The FNAL ROC room is equipped with a dozen Linux PCs with multiple LCD displays, fast network connections, a three terabyte (TB) file server, a web server, high quality video conference capability, a web camera, etc. More details of the FNAL ROC room infrastructure can be found from the FNAL ROC home page (http://uscms.org/roc).

The main focus of the FNAL ROC group activities is independent of location. We concentrate on working closely with our CMS colleagues in the areas of data acquisition, triggers, data monitoring, data transfer, software analysis, and database access, in order to commission the sub-detectors with the common goal of efficient and high quality data taking for physics. One of our goals is to make the relevant information necessary to do data quality monitoring available to the entire CMS collaboration. Web-Based Monitoring (WBM) [4] is an example of such a project we have started for this purpose. RunSummary [5] is one WBM tool already in extensive use by our CMS colleagues, including those at the Green Barracks, the temporary MTCC control room at SX5. We have used features of WBM as an integral part of the monitoring tools in MTCC-I.

1.2 Introduction to the FNAL ROC MTCC-I Involvement

From the FNAL ROC, we have been involved in CMS activities for some time. We have taken HCAL test-beam data quality shifts, catalogued HCAL calibration and test-beam data [6], run the event display using the the Interactive Graphics for User Analysis (IGUANA) visualization toolkit [7, 8, 9, 10] in CMSSW [7, 11], and prepared general WBM tools such as RunSummary.

MTCC-I was the first time the global DAQ was taking data with more than one sub-detector in the readout. At SX5, hardware and software configurations were frequently changing during most of the month of August. The exception was the last weekend when we had a short period of reasonable stability. The global data quality monitoring (DQM) processes were not running at SX5 in a consistent manner that permitted us to connect and view the results in real-time. Under such conditions and with strict time constraints, we decided to go with the following strategy:

- Get all of the global DAQ MTCC data transferred as fast as possible using the T0/T1 data transfer facility;
- Automate the process to systematically run the IGUANA Event Display and as many existing DQM programs as possible;

- Make all of the DQM results available and easily viewable from anywhere through the WBM tools;
- Communicate with our colleagues at CERN through video and phone conferencing, logbook entries and frequent emails;
- Take real-time DQM shifts as needed.

2 Data Transfer

During MTCC-I, FNAL T1 performed the data transfer from CERN T0, archived the data in dCache/Enstore ([12], [13]) and distributed the data to several USCMS university sites. We utilized the Physics Experimental Data Export (PhEDEx) [14] with the Storage Resource Management (SRM) [15] back end to transfer the MTCC data from CERN T0 to Fermilab dCache. The FNAL ROC played a major role during quasi-online data monitoring and offline MTCC data analysis, putting special requirements on the data transfer from CERN T0 to FNAL T1.

2.1 Overview

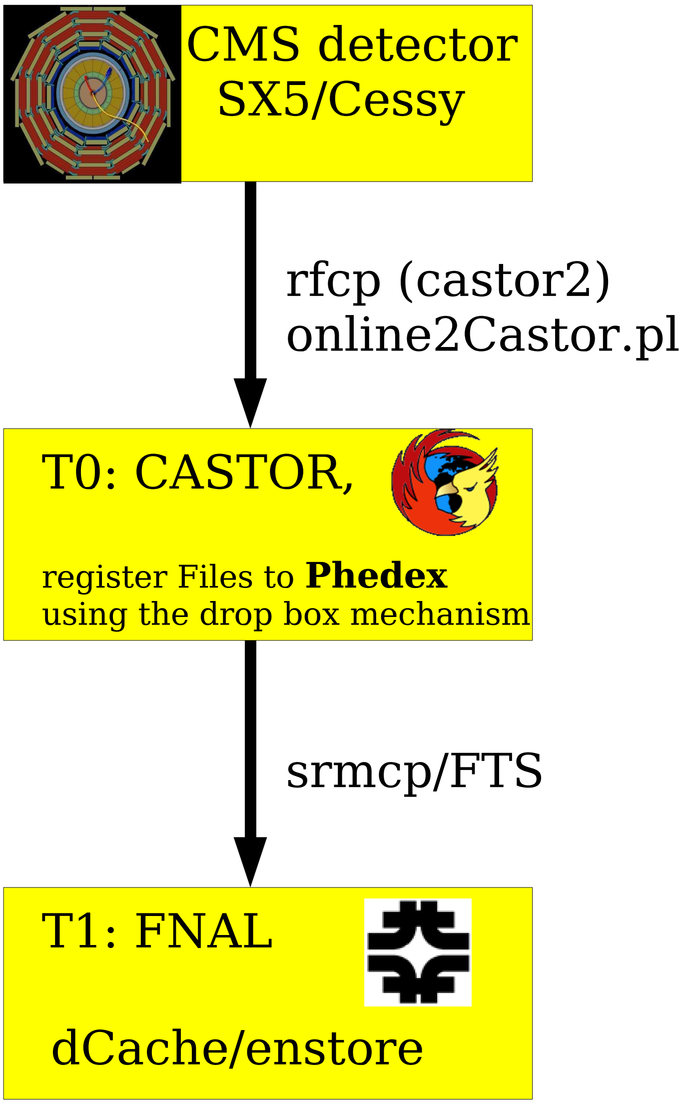


Figure 1: MTCC data transfer from the online DAQ machine to T0/CASTOR to T1/FNAL.

Figure 1 shows the steps, programs and file protocols involved in transferring MTCC data from the online DAQ machines to the CERN Advanced STORage manager (CASTOR) [16]) and then to FNAL. We have the following steps in order to move the data from the online area to offline sites like FNAL:

- Transfer the files from online to CASTOR;
- Inject the file information into the PhEDEx Database (TMDB) using the drop box mechanism;
- Remote site PhEDEx agents query TMDB and download the files that are ready to be transferred using Storage Resource Management Copy (srmcp) client or File Transfer Service (FTS) [17]. At Fermilab we used srmcp to transfer the files from CASTOR to dCache;
- Then the files get migrated from dCache to the Enstore back-end tape system automatically. As soon as files are in dCache, they are accessible independently of the file having moved to tape or not.

2.2 Transfer at T0: (SX5 to CASTOR)

The script online2CASTOR.pl was used to transfer the data from Cessy to Meyrin. The data files are directly copied to CASTOR using rfcp. The files are then registered to the PhEDEx dropbox. The dropbox is basically a xml file that is evaluated by the PhEDEx drop box agent which then triggers the injection into the PhEDEx data base and the transfer to Fermilab.

Delays can happen in both processes. For example, the frequency in which files are being sent to CASTOR after the Storage Manager [18] closes a file depends on the network connection between Cessy and Meyrin. It is estimated that most of the files arrive at CASTOR within half of an hour. Any error with the dropbox and PhEDEx agents can cause the injection into PhEDEx to fail.

2.3 Transfer from T0 to T1 Site

The transfer of data from T0 to T1 is handled by PhEDEx through CASTOR at CERN and dCache at FNAL. The basic procedure involves:

1. The files in the dropbox are injected into TMDB for remote sites to query and download;
2. PhEDEx export agents at CERN upload the necessary information for remote sites to download the requested files;
3. If possible, the export agents also pre-stage the files from tape to cached disk in CASTOR. This requires good interaction between PhEDEx client, CASTOR stager, CASTOR tape system and cached disk;
4. At FNAL, we have download agents running to query TMDB and initiate the SRM third-party transfer from CASTOR to dCache if files are available for transfer.

The smooth transfer from offline T0 to T1 site requires all of the components to work coherently and smoothly. Any disruption in the injection process, CASTOR or dCache will cause a delay of the transfer.

2.4 The Transfer Performance and Bottlenecks

The files are usually transferred from the online machine to CASTOR within half of an hour. We implemented the policy that the most up to date data was transferred first (Last-in, First-Out, or, LIFO). This had the consequence that some files were left behind when transfers were delayed and took significantly longer to be transferred. As we do not have access to the exact time when files were written to tape by the Storage Manager, the estimated time is based on the start and stop time values available in the RunSummary database. From CASTOR to dCache, the fastest files were able to arrive in about 8 minutes; about 46% of runs had some files arriving at Fermilab within one hour during the ramping up to 4 Tesla. The overwhelming majority of files arrived in dCache after one to two days, or even longer.

As MTCC-I took place during the summer, many experts including those working on data transfer - dCache/SRM, CASTOR and PhEDEx - were either on vacation or attending conferences. This affected our ability to fix problems quickly as well as push the system to the maximum performance. Other factors that affected the transfer performance were:

1. CASTOR had some problem with its stager query, and would unpredictably not return the status of files available in its cached disks. For some time during MTCC-I, PhEDEx had to run a faked stager which claimed all the files were in the cached disk. This caused quite a lot of transfers to fail, and those on tape could not be ready for transfer. CASTOR developers knew of this problem, and had been actively working on this since July, but did not have a resolution for the first days of MTCC-I.
2. PhEDEx: For a few days, the injection of new data into PhEDEx failed and refused to start. The problem was only solved after the intervention of Tony Wildish.
3. dCache: dCache/SRM also had some problem during the summer (e.g., one down pool can stall all transfers) while experts were away. This also prevented us from achieving high transfer throughput.

Other factors like the transfer policy and bandwidth sharing also caused the large average transfer latency.

2.5 Future Plan and Improvement for MTCC Phase II

For MTCC-II, the situation should be better. dCache was upgraded which will prevent one pool from stalling transfers. In addition the monitoring of the dCache pools was improved. Our experts are standing by to help. Developers also upgraded CASTOR before the start of MTCC-II. We have gained a lot of experience during the MTCC-I, so the issues related to the PhEDEx injection should be resolved. There are some ideas to further reduce the transfer latency in order to achieve real quasi-online monitoring. We may need to explore the possibility to transfer files out of the CASTOR cached disk before putting them on tape. Otherwise, the interaction with the CASTOR tape system will always be a bottleneck.

3 Automation and Bookkeeping

After some discussions with the CMS Data Management project leader, Lee Lueking, in the spring of 2006, we realized that the official CMS Data Set Bookkeeping System (DBS) [7, 19] would not be ready for MTCC-I. Although the data would be transferred to dCache at FNAL, without any database, bookkeeping tools, processing summary, etc. we would not know when the data arrived, what sub-detectors and triggers were in the data, and the explicit location (path in dCache) of the data.

In this section, we describe our solution to the data bookkeeping for MTCC-I at the FNAL ROC, based in part on the Quasi-Online Processing Summary at CERN [20] and the RunSummary tools [5]. In addition, we provide details on our automation scheme for job submission for the file conversions and data quality monitoring.

3.1 Job Submission and Process Monitoring

Since the FNAL ROC became involved in the MTCC-I monitoring on short notice, the scripts to control the data processing were developed as the data files arrived. We decided to process each subsystem independently instead of combining all of the subsystems into one executable. Each local subsystem expert was responsible for the analysis code, the executable, the configuration files, setting up the correct runtime environment, making sure that the necessary calibration files were available, etc. This allowed us to run different code versions and apply subsystem specific patches if necessary. Problems in one subsystem did not affect the processing of the other subsystems, and the processing could be distributed throughout the batch system. However multiple access to the same files can be a disadvantage.

Figure 2 illustrates the different processes used to analyze the data files and to make the histogram results available with the WBM tools. We processed each file separately as they appeared in dCache. All of the histograms from the available file fragments which belong to a given run were combined whenever a new file was processed by any DQM subsystem. This allowed us to make the histograms available on the web even when all of the file fragments associated to a run had not yet arrived or if the data taking of the run was still ongoing.

The processing scripts were written in Python [21]. The main script creates the configuration (parameter) files, the cmsRun scripts, the Condor [22] job description files, the Condor DAG files to process the jobs and to run the histogram merger.

A cron job [23] ran every fifteen minutes and checked if new data files had arrived at Fermilab. A second cron job on a different fifteen minute cycle queried the RunSummary database to get a list of runs per subsystem. Each new

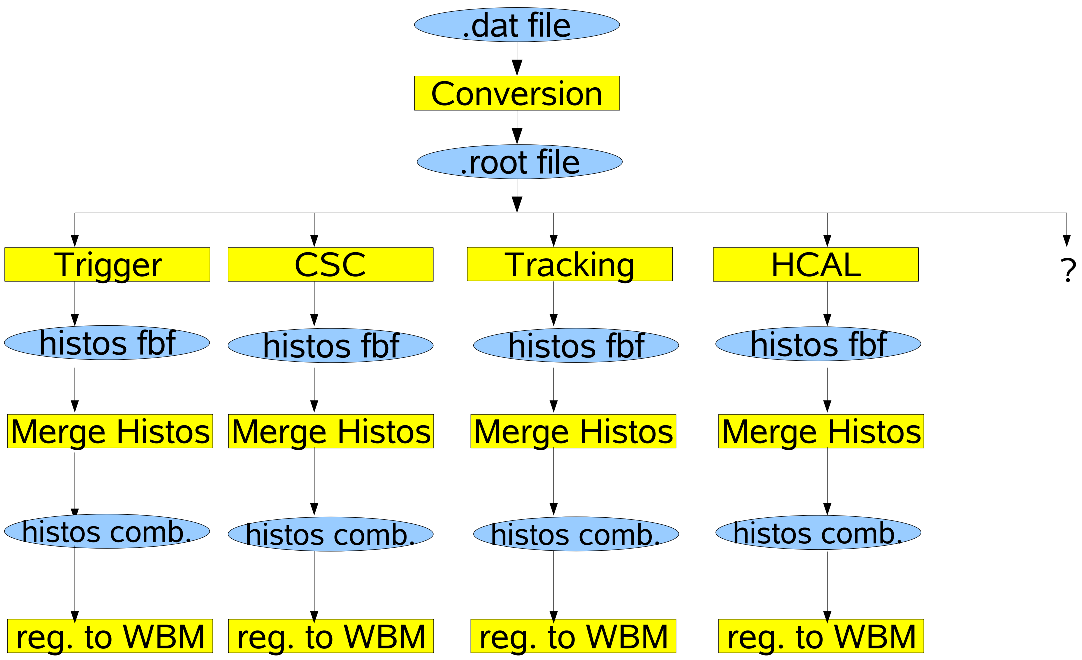


Figure 2: FNAL ROC data processing flow chart.

file was first converted into a ROOT [24] file, and then for any detector subsystem which was in the global readout, the DQM process was started.

We integrated different available tools:

- Mass storage dCache. The MTCC-I data consists of 1.7 TB, 7869 files of converted ROOT files in dCache and 7.1 TB of raw .dat files.
- Batch system (Condor, DAGman [22]). We used the Condor-based work cluster at Fermilab for all of the processing. We used a common group account "cmsroc", and to make sure that we could process the data as it arrived, "cmsroc" was given high priority. Between 17 August 2006 and 7 September 2006, we used 12575 CPU hours on the Fermilab Condor farm. We used the Condor DAGman facility to synchronize the processes running file by file. DAGman ensured that the process to merge the histograms started only when the processing for all new files associated to a DAQ run were finished.
- ELog [25]. The electronic logbook was used in the traditional sense as a logbook. In addition, the DAGman processes automatically made entries in the logbook whenever new merged files were completed. We did not yet use the feature to document errors, but it is an easy extension of the current system.
- WBM. The Web-Based Monitoring tools allowed us to browse the histograms and other run related information via the internet. The merged histograms were copied to an agreed upon directory where they were automatically picked up by WBM (see also Section 3.7). The code and the processed histograms add up to 38 GB on the shared file system.
- CVS [26]. The jobs scripts, templates and configuration files are in the LPC CVS repository.

3.2 File Conversion

The data files were shipped in the streamer format (.dat), causing a few problems. The size of the .dat files were approximately four times as large as the corresponding files in the ROOT format which makes use of data compression. Also, the CMSSW framework could not read this .dat format directly from dCache, therefore files had to be staged to disk before they could be processed.

For the above reasons we decided to run a conversion routine which converted all the incoming data files from the .dat format to the ROOT format. The ROOT files were approximately four times smaller and could be accessed in dCache directly. All subsequent processes depended on having the conversion done first.

3.3 Failure Modes

We observed various failure modes that will have to be addressed. After looking at the log files, the modes of failure can be categorized in four ways.

1. No log files are written. We will work on a scheme to archive the log files so we can scan them for errors.
2. An input/output error was reported, so we should be able to catch this in the future.
3. There are no errors in the log file, so everything looks normal.
4. The log files terminate before the datagram congestion control protocol (dccp) [27] command which copies the files from dCache is executed.

In the first two failure modes, we should be able to improve our fail-safes. For the other failure modes, further consideration is needed, as we are not able to reliably reproduce the failures.

3.4 Archiving

A script is used to back up the log files, the job scripts, and the configuration files that were used in processing the MTCC data. The files in the sub-directories in /uscmst1b_scratch/lpc1/cmsroc/MTCC/DQM/XXX/YYY where XXX = convert, hcal, muon, tracker and trigger and YYY = cfg, logs, condor, scripts and dag are backed up to tape.

Any files that have not changed since the last backup are added to the tar archives: cfg.tgz, condor.tgz, dag.tgz, logs.tgz, and scripts.tgz. We only keep files on disk for two days. This script is executed by a cron job once a day.

3.5 Remaining Issues for MTCC-II

Although not exhaustive, the following list is a summary of the major issues with the job submission and bookkeeping from MTCC-I which remain to be resolved for MTCC-II:

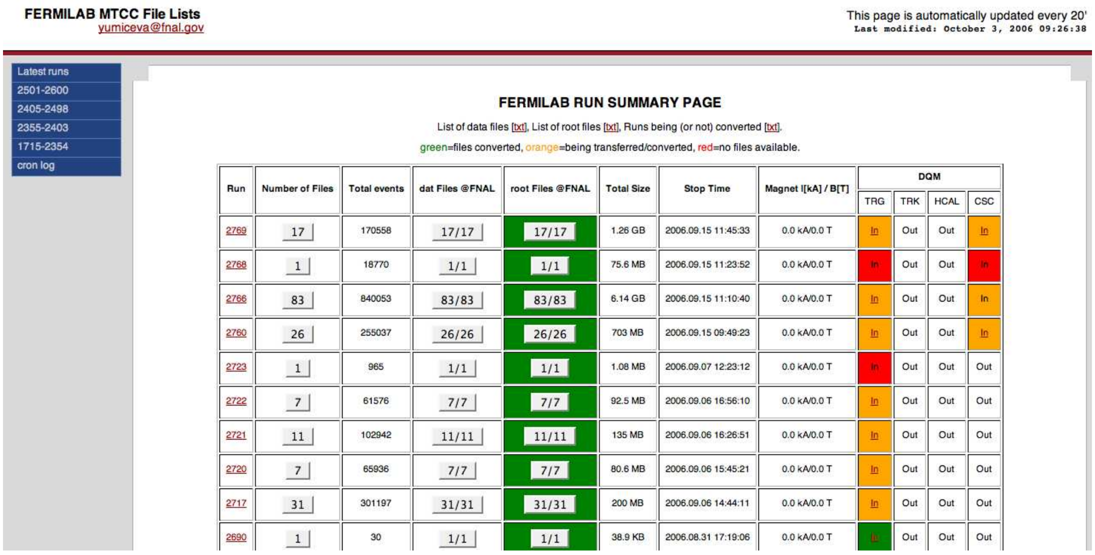
- Better error handling. Jobs should recover automatically.
- Better handling of log files. We need to improve the tools to archive and retrieve the log files. Also it would be nice to have tools to automatically scan log files for any error that occurred.
- Send processing errors directly to the logbook (or an error logger). The message should describe what went wrong and provide instructions for the shifters.
- Need to manage run-dependent parameters like configuration, geometry, calibration, cable map files etc. Many of this things reside in different files and need to be maintained by the expert.
- Use a small database for bookkeeping and to make reports.
- Need to reduce the process latency. The conversion of the files should be triggered as soon as new files arrive in dCache instead of having a cron job checking for the new files every fifteen minutes. The start of all monitoring processes should start as soon as the file conversion is done.

3.6 Process Summary Page

The FNAL ROC process summary web page (Figure 3) contains information about the status of the MTCC data. On the top right side, the date of the last update of the page is shown. On the left side, there is a navigation bar to older pages and to the log file of the cron job.

The summary pages have a table sorted by run number with the following information:

- *Run Number:* Ordered by the most recent run recorded at SX5 and with at least one file stored in CASTOR at CERN.



FERMILAB MTCC File Lists
yumiceva@fnal.gov

This page is automatically updated every 20'
Last modified: October 3, 2006 09:26:38

Latest runs
2501-2600
2405-2498
2355-2403
1715-2354
cron log

FERMILAB RUN SUMMARY PAGE

List of data files [txt], List of root files [txt], Runs being (or not) converted [txt].

green=files converted, orange=being transferred/converted, red=no files available.

Run	Number of Files	Total events	dat Files @FNAL	root Files @FNAL	Total Size	Stop Time	Magnet I[kA] / B[T]	DQM			
								TRG	TRK	HCAL	CSC
2769	17	170558	17/17	17/17	1.26 GB	2006.09.15 11:45:33	0.0 kA/0.0 T	In	Out	Out	In
2768	1	18770	1/1	1/1	75.6 MB	2006.09.15 11:23:52	0.0 kA/0.0 T	In	Out	Out	In
2766	83	840053	83/83	83/83	6.14 GB	2006.09.15 11:10:40	0.0 kA/0.0 T	In	Out	Out	In
2760	26	255037	26/26	26/26	703 MB	2006.09.15 09:49:23	0.0 kA/0.0 T	In	Out	Out	In
2723	1	965	1/1	1/1	1.08 MB	2006.09.07 12:23:12	0.0 kA/0.0 T	In	Out	Out	Out
2722	7	61576	7/7	7/7	92.5 MB	2006.09.06 16:56:10	0.0 kA/0.0 T	In	Out	Out	Out
2721	11	102942	11/11	11/11	135 MB	2006.09.06 16:26:51	0.0 kA/0.0 T	In	Out	Out	Out
2720	7	65936	7/7	7/7	80.6 MB	2006.09.06 15:45:21	0.0 kA/0.0 T	In	Out	Out	Out
2717	31	301197	31/31	31/31	200 MB	2006.09.06 14:44:11	0.0 kA/0.0 T	In	Out	Out	Out
2690	1	30	1/1	1/1	38.9 KB	2006.08.31 17:19:06	0.0 kA/0.0 T	In	Out	Out	Out

Figure 3: FNAL ROC process summary page (http://nippon.fnal.gov/lpc1/cmsroc/MTCC/check_mtcc).

- *Number of Files:* Number of files per run in CASTOR. We provide a link with a list of files with their full path.
- *Total Events:* Total number of events per run. This value is queried from the RunSummary database.
- *Data Files at FNAL:* Total number of data files per run copied to FNAL dCache. A link per run number is provided with a list of the data files including the explicit paths.
- *ROOT Files at FNAL:* Total number of ROOT files per run at FNAL dCache. The ROOT files are obtained by converting the data files from the .dat format to the ROOT format. A script automatically checks that a converted file for each data file exists. Otherwise, the run number is added to a list that will be used to submit jobs for conversion. The cell can have three colors: *red* if there are zero ROOT files available for a run, *orange* if some non-zero fraction of the total number of files for a run, and *green* if all of the files for a run are available. See Figure 4 as an example.
- *Total Size:* Total size of the ROOT files in FNAL dCache per run number.
- *Stop Time:* Run stop time. This value is queried from the RunSummary database.
- *Magnet I [kA]/B[T]:* Average magnet current in kA and the magnetic field in Tesla.
- *DQM Status:* These columns contain the information about the DAQ status per run for each subdetector: trigger (TRG), tracking (TRK), hadronic calorimeter (HCAL), and muon cathode strip chamber (CSC). There are two options: "In" when a subdetector has been included in the run, and "Out" when the subdetector was taken out of the DAQ, in which case the color of the cell in the page will be *white*. When the subdetector was included in the run, the cell can have a *red* color if the DQM plots have not been processed for that run, an *orange* color if the DQM plots are available for a non-zero fraction of the run, and *green* color if the DQM plots are available for all of the events in a run. In the last two cases, a link to the DQM plots is provided on the page.

The Python script to produce the page can be found in the LPC CVS repository:

```
setenv CVSROOT lpccvs@cdcvs.fnal.gov:/cvs/lpc
cvs co MTCC/scripts/check_mtcc.
```

This script can take three arguments:

```
./check_mtcc.py [<range of runs>] [<convert>] [<check ''test'' directory>].
```

The first argument is a string with a range of runs. The second argument enables or disables the conversion of the .dat files to the ROOT format. The third argument enables or disables the scan of the ROOT files stored in the "test" directory. For example:

2610	14	65991	14/14	14/14	3.15 GB	2006.08.27 21:46:05	18.2 kA/3.8 T	In	In	In	In
2605	29	139750	29/29	29/29	6.40 GB	2006.08.27 20:59:23	18.2 kA/3.8 T	In	In	In	Out
2603	79	379296	79/79	79/79	18.3 GB	2006.08.27 20:18:03	18.2 kA/3.8 T	In	In	In	In
2602	206	1000502	206/206	206/206	48.6 GB	2006.08.27 19:32:07	18.2 kA/3.8 T	In	In	In	In
2601	90	462820	90/90	90/90	20.6 GB	2006.08.27 17:50:46	18.2 kA/3.8 T	In	Out	In	In
2587	3	863688	3/3	3/3	718 MB	2006.08.27 07:41:42	18.2 kA/3.8 T	In	In	In	In
2584	8	675063	7/8	7/8	1.64 GB	2006.08.27 05:13:49	18.2 kA/3.8 T	In	In	In	In

Figure 4: Enlarged view of the tables from the FNAL ROC process summary web page.

`./check_mtcc.py ''-2501-2600'' 1 0.`

A cron job is run on the User Analysis Farm (UAF) [28] farm by the user `cmsroc`. The script is run every fifteen minutes on the latest runs and every hour on the older runs. The cron job has been installed on the worker node `cmswn052.fnal.gov`.

3.7 RunSummary

The Online Monitoring Database Service (omds) Oracle database at P5 contains an ever expanding wealth of information related to the status and history of CMS. In particular, this database contains run and trigger information spread across various database schemas and table. The database is located behind a firewall and is not directly accessible outside P5. To facilitate access to this information by remote institutions, including but not limited to the FNAL ROC, we have created a RunSummary servlet [5] running on a general user machine located on the firewall border. The servlet uses Apache Tomcat and Jakarta [29, 30] technology in conjunction with an Apache web server running on the same machine. The servlet is available to all outside users.

The primary entry point for the RunSummary servlet is a query form shown in Figure 5. The form allows the user to search for runs satisfying various optional criteria, including magnet current, events triggered, date and time, triggers included, etc. An example of a multi-run result from this page is shown in Figure 6; the information is also available in text-only format for the convenience of script writers. In Figure 7, the RunSummary for a single run is queried. Each of the links can provide more detailed information about the quantity displayed, usually a plot of the quantity as a function of time during the run. Also shown in Figure 7 is the plot of the magnet current as a function of time for Run 2241.

For MTCC-II, as outlined in CMS note [4], the functionality of the RunSummary and related pages will be expanded to include data from the different subdetectors. An example of future plans can be found off the Web-Based Monitoring home page, at the Magnet History link:

`http://cmsdaq.cern.ch/cmsmon/`

4 Data Quality Monitoring

Data quality monitoring (DQM) provides a homogeneous monitoring environment [31, 32] across various applications in the CMS experiment:

- The High-Level Trigger (HLT) filter farm to guarantee the quality of the real-time data taking;
- The local DAQ of the various subdetector groups for data integrity and hardware performance checks;
- Validation jobs in a purely offline mode.

DQM includes a suite of tools:

- Tree-like directories with histograms, profiles, scalars and strings;

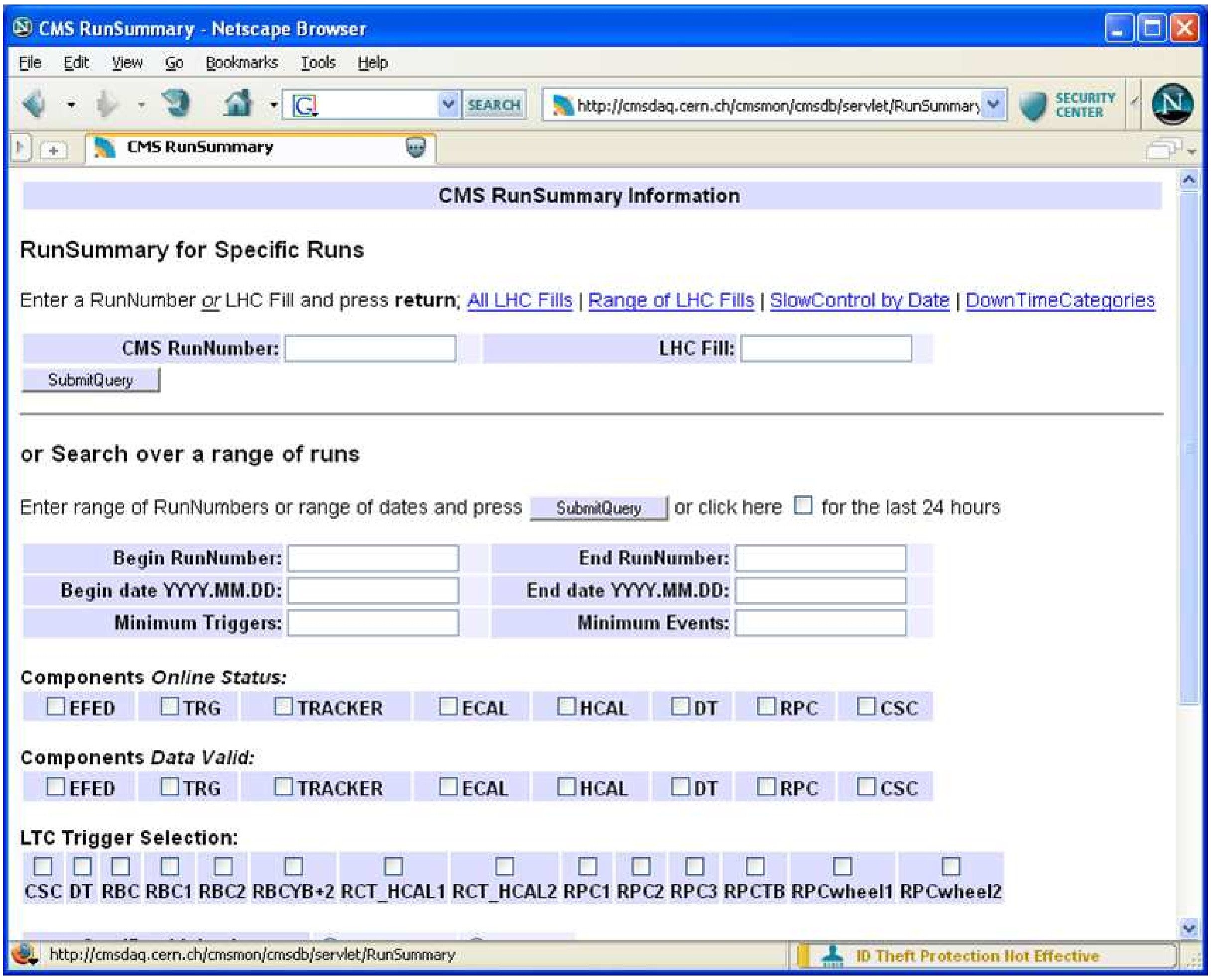


Figure 5: The RunSummary query page.

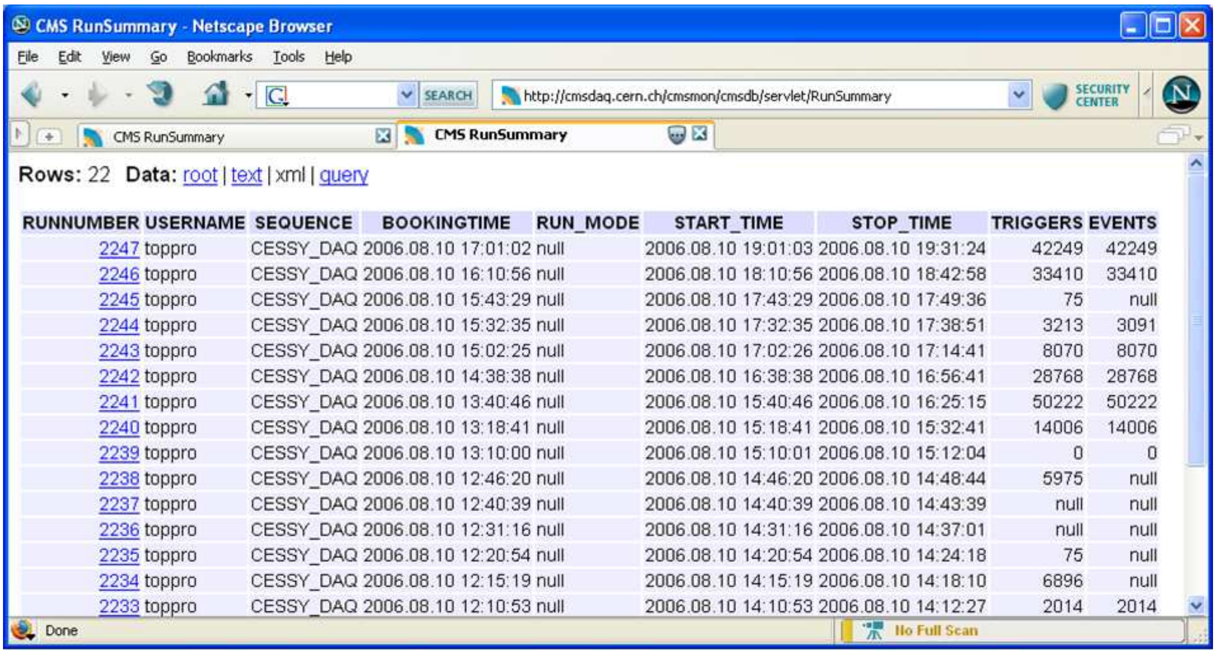


Figure 6: RunSummary query results for multiple runs.

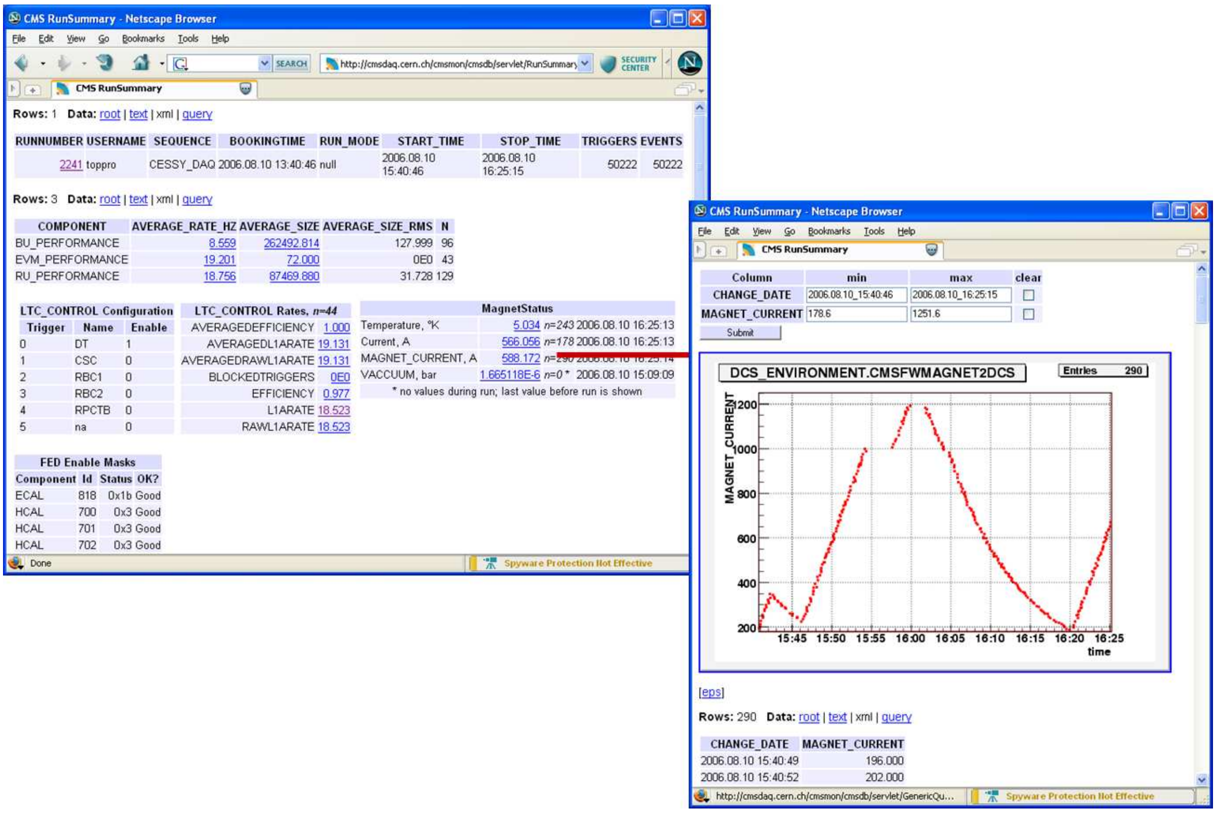


Figure 7: RunSummary query for Run 2241 with the magnet current as a function of time.

- Quality tests that can produce warnings, errors and alarms;
- Visualization applications;
- The transfer of monitoring information to remote nodes.

The DQM framework is designed to manage sets of monitor elements, e.g. histograms, from the creation in monitoring producers (sources), to the organization and redistribution on a periodic basis (collectors), to their final use by the monitoring information consumers (clients).

Sources are defined as individual nodes that have direct access to the interesting information, e.g. the HLT histograms. The creation and update of monitor elements at the source can be the result of processing input event data (event consumers) or input monitor elements (monitor consumers).

Sources are only connected to one collector, although one collector can be connected to multiple sources. All source-client communication is carried out through the collector(s).

Clients obtain a list of available monitoring information (monitorables) from all sources combined from the collector. The clients can subscribe to and receive periodic updates to any desired subset of the monitorables. Client applications should be customized for each subsystem.

4.1 Trigger

The trigger for MTCC-I consisted of an Local Trigger Control (LTC) module OR-ing the LTC's of muon detector elements and sending six trigger bits as data. The most common trigger bit assignment is listed below:

- bit 0 : DT
- bit 1 : CSC
- bit 2 : RBC1 (RPC technical trigger for wheel +1)

- bit 3 : RBC2 (RPC technical trigger for wheel +2)
- bit 4 : RPC-TB (RPC trigger from the trigger board)
- bit 5 : not used.

The bit assignment varied throughout MTCC-I as different trigger configurations were experimented with; in order to interpret these bits it was necessary to consult the MTCC shift or trigger ELog entries for the run(s) considered to confirm the specific trigger bit assignment. For most of the MTCC-I runs, the trigger configuration was stored in the CDAQ_CONDITIONS.LTC_CONTROL database table. In order to view the contents for a particular run, one used the RunSummary web page. Two columns specified the trigger: both the TriggerName (DT, CSC, etc.) and the TriggerEnable flag (0 or 1). Also, it was possible that the trigger configuration was changed during a run, and in this case multiple configurations would appear in the RunSummary page.

Data quality monitoring software consisted of an LTC package compiled in CMSSW_0_9_0 which converted the six bits plus event header data to a simple ROOT tree. The leaves of the tree included the event header data and a six bit mask showing the active trigger bits:

- run: run number
- event: event number
- triggerNumber: trigger number
- mask: six bit trigger mask (typically encoded as listed above)
- gpstime: GPS time stamp (from the network and only updated typically every 3 seconds or so)
- orbit: beam orbit number
- bunch: bunch crossing number
- inhibit: number of inhibited triggers at that point in the run

In addition, the LTC analyzer produced a histogram of the trigger accepts for each bit (mask) and a 2D histogram of the 6X6 trigger bit coincidences (overlaps). More information on the trigger configuration and meaning of the signals can be found at:

`https://twiki.cern.ch/twiki/bin/view/CMS/OnlineWBTrigger`.

The software was installed as follows. Obtain the (currently private) LTC package from Peter Wittich or Jeff Berryhill:

```
# simple analyzer code from Peter
cp -r ~berryhil/CMSSW_0_9_0/src/LTC
cp ~berryhil/CMSSW_0_9_0/src/makeLTC.cfg
cp ~berryhil/CMSSW_0_9_0/src/do_run_fnal.sh
```

Compile and run the LTC package by editing the `do_run_fnal.sh` shell script to find `.root` files in the appropriate location and then execute the shell script:

```
sh do_run_fnal.sh run_number largest_frag_number
```

where the first and second arguments are the run number and the number of files for that run to be merged into a single output tree.

The `.root` output can be browsed online for runs processed by FNAL, CERN, or in some cases from an LTC decoder running directly in the HLT executable online. Figure 8 shows an example of WBM output for the FNAL ROC processing: the trigger bit histogram ("mask") is presented for MTCC run 2589.

Plans for future development for MTCC-II include: decoding data from the global level 1 trigger, to be newly implemented; decoding trigger primitives data; and including these decoding and histogramming features into an online DQM client, as was done near the end of MTCC-I for the LTC. Also, the online DQM client subscribing to the HLT LTC DQM objects should be placed under the control of a DQM Function Manager so that it starts and stops reliably at the beginning and end of runs. The user and location of running online DQM processes must be worked out for MTCC-II.

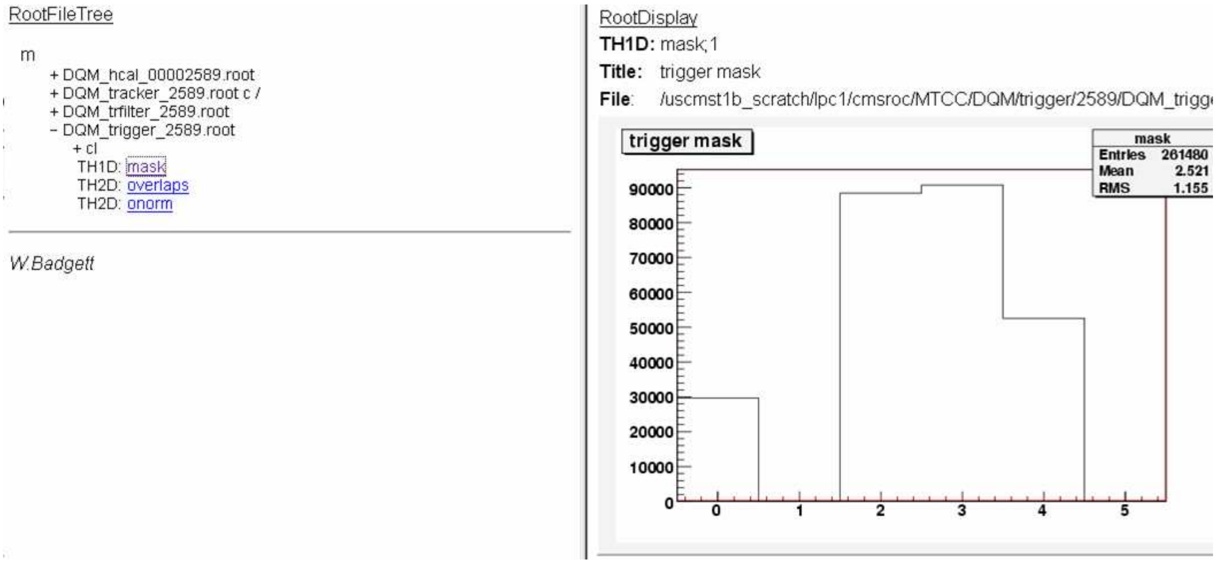


Figure 8: Example of the WBM output for LTC data for run 2589 as processed at the FNAL ROC.

4.2 Tracker DQM

All of the data files of the MTCC-I running period (August 24 - August 29) have been processed by the tracker DQM software, which consists of two steps: *(1) reconstruction of clusters* and *(2) filter and full reconstruction*. The DQM output of the first step results in histograms for each detector module, which are stored in a ROOT file in a directory structure corresponding to the tracker geometry. The second step filters events based on the reconstructed clusters and performs a track reconstruction using the cosmic track finder. The corresponding DQM output displays various track variables and is stored in the ROOT histogram file. The two reconstruction steps are described in more detail in Sections 4.2.2 and 4.2.3. The data processing Python script runs both steps automatically, and a more detailed description is given in Section 4.2.1.

4.2.1 Software Framework

For MTCC-I, the tracker DQM software in release CMSSW_0_9_0 has been used. The following patches had to be applied to this version:

- `EventFilter/SiStripChannelChargeFilter V01-01-02`
- `RecoTracker/SingleTrackPattern V01-02-01`
- `EventFilter/SiStripRawToDigi V01-03-02`
- `DQM/SiStripMonitorDigi V02-03-00`
- `DQM/SiStripMonitorCluster V02-03-00`
- `DQM/TrackerMonitorTrack V02-03-00`

The Python script needed to process the data files as well as all necessary database and configuration files were accessible from the LPC CVS repository by checking out the MTCC/tracker package.

The processing of data files is invoked by:

```
./process_tracker_fbf.py <run_number> <magnetic_field_value> <batch_submission>
```

The script used the following arguments:

1. <run_number> : run number which should be processed
2. <magnetic_field_value> : average magnetic field value for the run in Tesla

3. <batch_submission> : enable or disable running on the CONDOR farm (y/n)

The script processed all available data files for the given run on a file-by-file basis. The data file was skipped if it had been previously processed. The output DQM files for each run was merged and published using the WBM tools.

4.2.2 Reconstruction of Clusters

The reconstruction of silicon strip clusters was controlled by the configuration file tracker_template.cfg, which can be found in the directory MTCC/tracker/Templates. The configuration file contained the hard-coded location of the FED cabling map as well as the pedestal and noise files. All these configuration files were provided in the form of a SQLite database files and stored in MTCC/tracker/db_files. The cluster reconstruction retained only clusters which passed the thresholds 4/3/5 for seed, channel and cluster respectively.

For each processed file, the cluster reconstruction resulted in two different kind of output files:

1. `DQM_tracker`: A simple ROOT file that contained the DQM histograms for the processed data file.
2. `POOL_tracker`: A pool reco file that contained the tracker clusters for all events in the processed data files. No data from other subdetectors was retained, and no filtering was done.

The POOL_tracker output file was used as input to the filter and reconstruction step, which is described in the following section.

4.2.3 Filter and Full Reconstruction

The event filter and full track reconstruction was controlled by the configuration file track_template.cfg2, which can be found in the directory MTCC/tracker/Templates. It filters the events based on the charge of the reconstructed clusters and peforms further reconstruction using the cosmic track finder. As a requirement for the input file, a cluster collection had to be present, therefore the output pool reco file from the above described cluster reconstruction step is needed as input.

The event filtering wass performed in two steps:

1. In the first step, clusters were only accepted if they were above certain thresholds. In ADC counts, the default thresholds for the Tracker Inner Barrer (TIB), Tracker Outer Barrel (TOB) and Tracker Endcap (TEC) are 25, 25 and 70 respectively. For the event to pass, there have to be at least three clusters in different layers of the tracker, whereas the layer counting corresponds to the standard layers of the tracker: 2 TIB layers and 2 TOB layers. The TEC is not counted.
2. In the second step, only events that have at least one reconstructed track are retained.

It is possible to define a vector of modules whose clusters are excluded from the filter decision. This vector contains the raw IDs of the modules (32 bit unsigned numbers) and is defined in the configuration file. Currently no modules are excluded and the above list is empty.

For each processed file, the full reconstruction results in two different kind of output files:

1. `DQM_trfilter`: A simple ROOT file that contains the DQM histograms for the processed data file. Only events with tracks are shown.
2. `POOL_trfilter`: A pool reco file that contains tracker clusters, rechits and tracks for all filtered events. No data from other subdetectors is retained.

An example DQM output is shown in Figure 9.

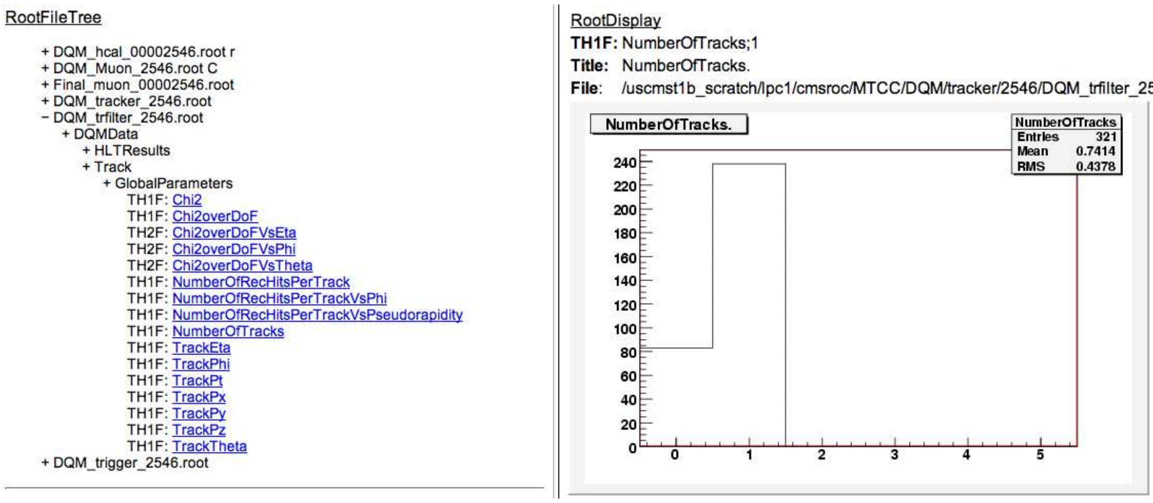


Figure 9: Example of the tracker DQM output for run 2545 (with $\mathrm{B} = 0$ T). The left column shows an expanded view the DQM result of the *filter and full reconstruction* step; the histogram on the right shows the number of tracks reconstructed by the cosmic track finder.

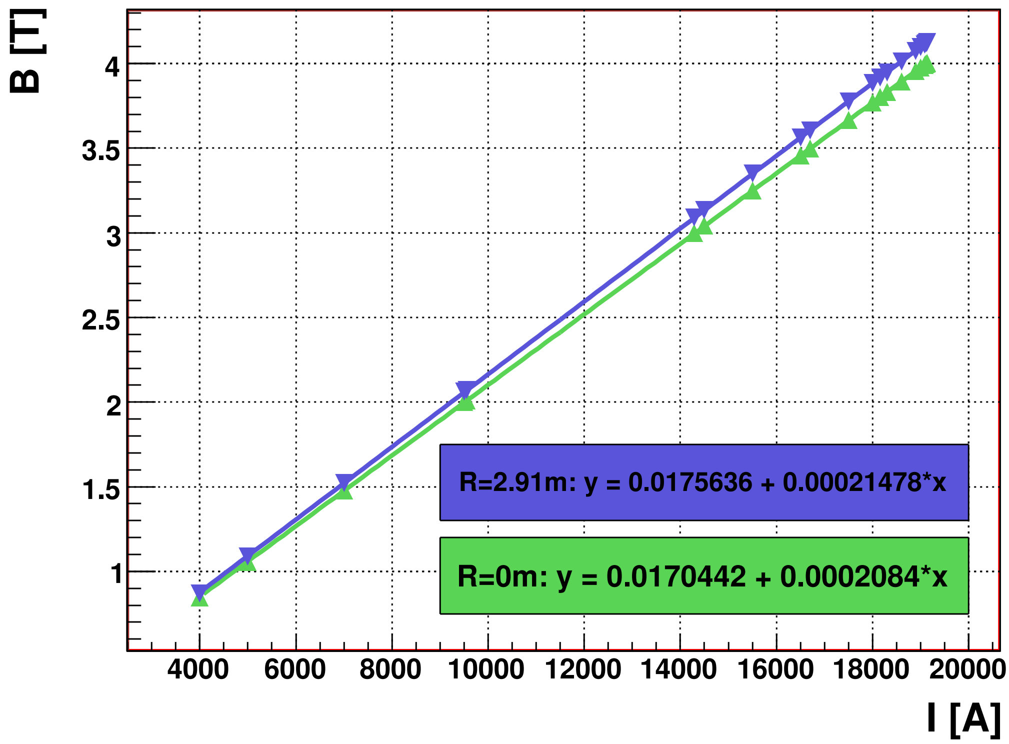


Figure 10: Magnetic field in Tesla as a function of the current. The B field values for $\mathrm{R} = 2.91$ m are the average of the magnetic field values as measured by the Hall probes. The values for $\mathrm{R} = 0$ m are calculated from the measured values.

4.2.4 Magnetic Field

During MTCC-I data runs were taken with the magnet turned off and with the magnet set to different currents, corresponding to different magnetic fields. Since a precise magentic field mapping is planned for MTCC-II, the current magnetic field values are obtained using four Hall probes located outside of the hadronic calorimeter barrel at $\pm 45^{\circ}$ (w.r.t. the horizontal plane). The measured values are used to calculate the magnetic field at $z = 0$. The results are shown in Figure 10 for $\mathrm{R} = 0$ m and $\mathrm{R} = 2.91$ m. The magnetic field is needed for the track reconstruction in order to calculate the transverse momentum of a track, and the magnetic field value is therefore given to the data processing script as a parameter (see Section 4.2.1). Since the magnet was ramped during the data taking period without stopping the run, there are a few data runs which are characterized by a varying magnetic field value. In these cases the average magnetic field is used to calculated the track $\mathrm{p_T}$.

4.3 Hadronic Calorimeter

The data quality monitoring for the hadronic calorimeter (HCAL) is performed in parallel for the HCAL barrel (HB), HCAL endcap (HE), HCAL outer (HO) barrel (outside the magnet), and HCAL forward (HF) subdetectors. For MTCC-I, the HF was not installed. The functionality for HCAL DQM is divided into two software packages:

- HcalMonitorModule: This package contains the analyzer which receives events from the framework and passes them to the individual monitors. Also included is a class which determines the monitors that analyze an event. For example, a pedestal monitor will only analyze pedestal events.
- HcalMonitorTasks: This package contains classes for each monitor type. Each monitor is instantiated and handled by the HcalMonitorModule class.

4.3.1 HCAL DQM

For MTCC-I, the release of CMSSW_0_9_0 was used with a special tag of HCAL DQM provided by the author, Wade Fisher. The HCAL DQM packages were built in the following subdirectory:

`/uscmst1b_scratch/lpc1/cmsroc/MTCC/software/CMSSW_0_9_0/src/DQM`

The Python script and configuration templates needed to process the data files can be checked out of the LPC CVS repository:

```
setenv CVSROOT lpccvs@cdcvs.fnal.gov:/cvs/lpc
cvs co MTCC/hcal
```

The processing of HCAL data files is invoked by the script:

`./process_hcal_fbf.py ''run_number'' ''batch_submission''`,

where the run number is a four digit integer, and the batch submission argument indicates whether to enable or disable running on the Condor farm with “y” or “n”.

The script processes all of the currently available data files for the given run on a file-by-file basis, meaning that each file belonging to the run is processed individually. The data file is skipped if it has previously been processed. The output DQM files for each run are merged and published using the WBM tools.

4.3.2 HCAL DQM Analysis

All of the MTCC-I HCAL DQM processing (CMSSW_0_9_0) used only one pedestal file (`peds_mtcc_2469.txt`) and one electronics map file (`emap_mtcc_hbheho_honew_aug15.txt`). No structure was in place for MTCC-I to make HCAL pedestal files readily available offline at the FNAL ROC. Instead, the pedestal and electronic map files were hard-coded into the configuration file as part of the HCAL DQM framework.

In addition, the threshold value in femto-Coulombs (fC) for an energy deposition of an HCAL cell, or digi, by a cosmic muon was set too low in the configuration file. This resulted in histograms with an order of magnitude of too many entries, and results that did not make sense.

For this note, MTCC run 2643 was reprocessed using the new release of CMSSW_1_1_0, an updated version of HCAL DQM, and the run-specific pedestal file accessed through the offline database. The electronics map file was also current. The HCAL digi thresholds were increased from the default value of 2 fC to a more reasonable value of 10 fC. The corrected MTCC-I HCAL DQM histograms for run 2643 can be accessed directly using the WBM tools.

`http://nippon.fnal.gov/cmsdb/servlet/DQMBrowser?RUN=2643.`

All of the MTCC-I runs will be reprocessed using the same HCAL DQM release, so that a complete and consistent set of histograms are available from the web.

4.3.3 HCAL DQM Histograms

The histograms produced by the HCAL DQM modules - DigiMonitor, MTCCMonitor, PedestalMonitor, etc. - use a common nomenclature that requires some clarification.

- *Bottom:* The wedge on the physical "bottom" of the CMS MTCC setup. It is in the negative Z direction with respect to the interaction region.
- *Top:* Opposite of bottom.
- *Yb1:* The yoke barrel section one mount point of the HCAL outer barrel module. The yoke is the magnetic flux return yoke.
- *Trigger Time:* The reconstructed time that all of the muon hits that create an energy deposit of at least 10 fC in a HCAL digi (cell). The x-axis is in units of bunch crossings, or buckets. The digi records 10 bunch crossings, so the histograms range from from 0 to 9.
- *Geo Occ Map:* The geometrical occupancy map in units of iEta (x-axis) and iPhi (y-axis) above a 10 fC threshold.
- *Elec Occ Map:* The electronics occupancy map in units of VME crate number (x-axis) and VME slot number (y-axis).
- *Hit Energy:* Distribution of reconstructed hit energies in fC.

The DigiMonitor geometrical occupancy maps are shown in Figure 11 for the HCAL barrel, endcap and outer detectors for Run 2643.

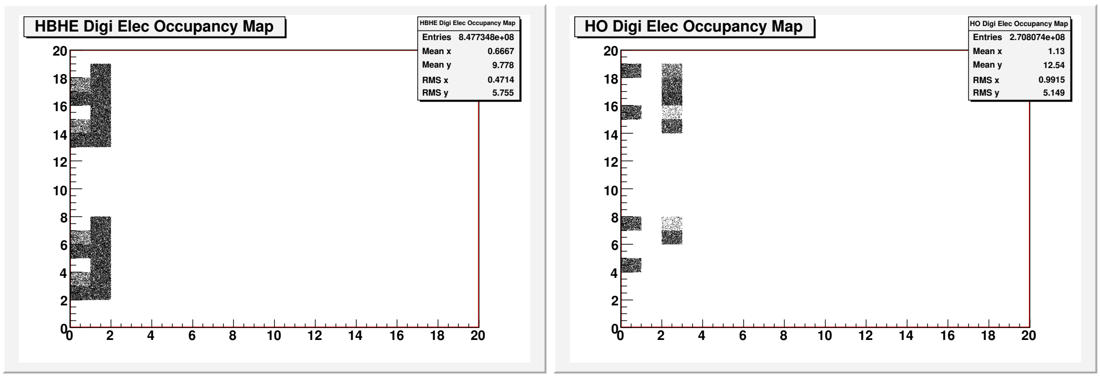


Figure 11: Run 2643, the electronics occupancy map in units of VME crate number (x-axis) and VME slot number (y-axis) (a) for the HCAL barrel and endcap detector and (b) for the HCAL outer barrel detector.

In Figures 12, 13, 14, 15, a common set of histograms from the MTCCMonitor are shown for different HCAL subdetectors: HCAL barrel bottom, HCAL barrel top, HCAL endcap and HCAL outer barrel.

4.4 Cathode Strip Chamber

The quasi-online CSC Muon DQM histograms displayed at the FNAL ROC, and linked from the RunSummary web page were based on the MTCC CSC analysis examples of M. Schmitt [33].

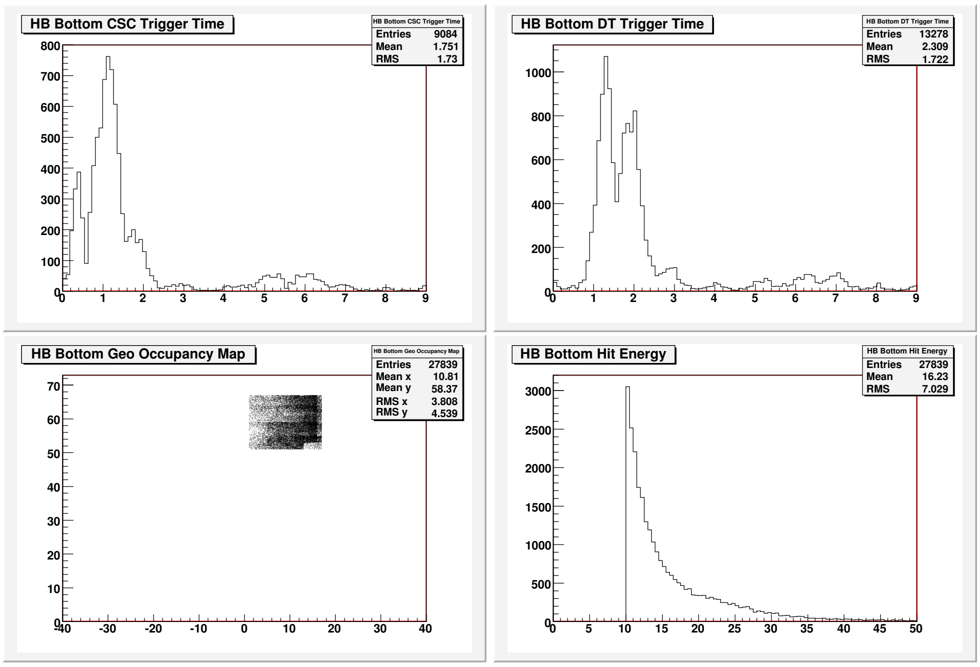


Figure 12: Run 2643, in the lower wedge of the HCAL barrel detector: (a) Reconstructed charge-weighted time of muon energy deposition in fractional units of bunch crossings. The events are obtained using a trigger in the cathode strip chambers of the muon system. (b) Reconstructed charge-weighted time of muon energy deposition in fractional units of bunch crossings. The events are obtained using a trigger in the drift tubes of the muon system. (c) Geometrical occupancy map in units of iEta (x-axis) and iPhi (y-axis) above a 10 fC threshold. (d) Distribution of reconstructed hit energies in fC.

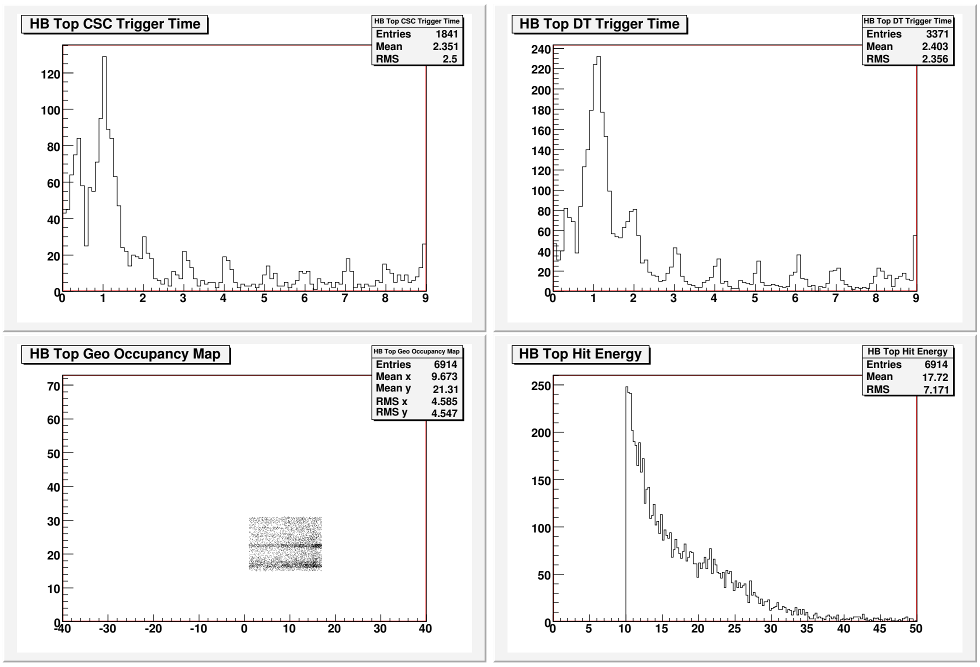


Figure 13: Run 2643, in the upper wedge of the HCAL barrel detector: (a) Reconstructed charge-weighted time of muon energy deposition in fractional units of bunch crossings. The events are obtained using a trigger in the cathode strip chambers of the muon system. (b) Reconstructed charge-weighted time of muon energy deposition in fractional units of bunch crossings. The events are obtained using a trigger in the drift tubes of the muon system. (c) Geometrical occupancy map in units of iEta (x-axis) and iPhi (y-axis) above a 10 fC threshold. (d) Distribution of reconstructed hit energies in fC.

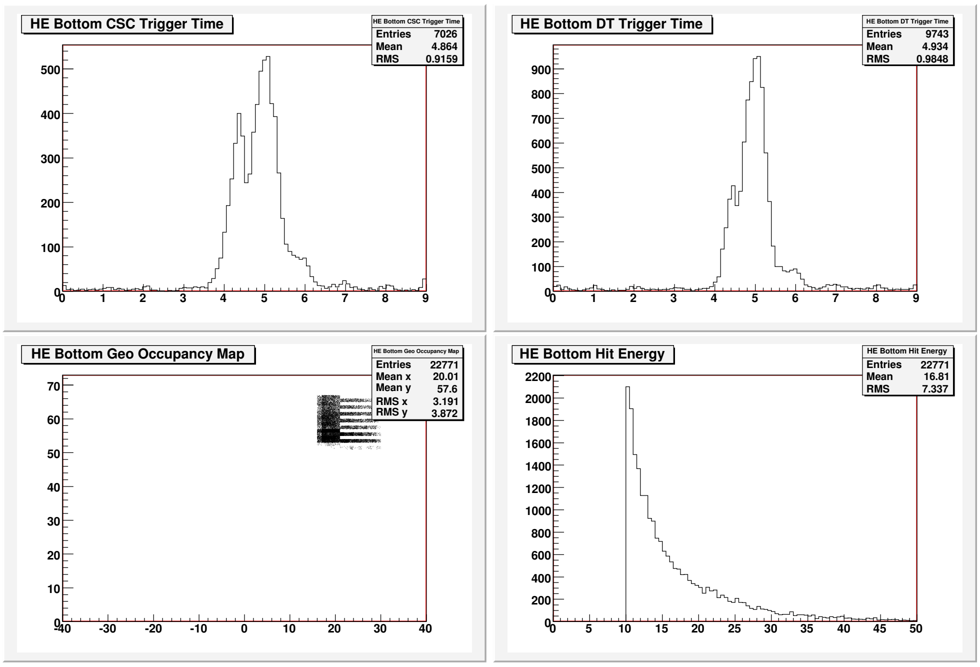


Figure 14: Run 2643, in the lower wedge of the HCAL barrel detector: (a) Reconstructed charge-weighted time of muon energy deposition in fractional units of bunch crossings. The events are obtained using a trigger in the cathode strip chambers of the muon system. (b) Reconstructed charge-weighted time of muon energy deposition in fractional units of bunch crossings. The events are obtained using a trigger in the drift tubes of the muon system. (c) Geometrical occupancy map in units of iEta (x-axis) and iPhi (y-axis) above a 10 fC threshold. (d) Distribution of reconstructed hit energies in fC.

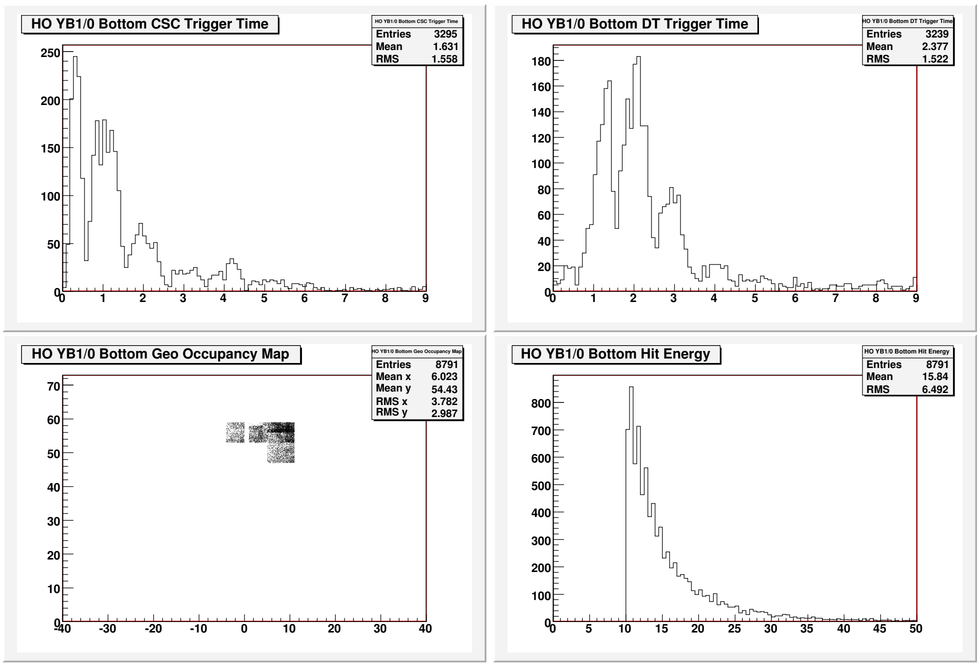


Figure 15: Run 2643, in the yoke barrel section one of the HCAL outer barrel detector: (a) Reconstructed charge-weighted time of muon energy deposition in fractional units of bunch crossings. The events are obtained using a trigger in the cathode strip chambers of the muon system. (b) Reconstructed charge-weighted time of muon energy deposition in fractional units of bunch crossings. The events are obtained using a trigger in the drift tubes of the muon system. (c) Geometrical occupancy map in units of iEta (x-axis) and iPhi (y-axis) above a 10 fC threshold. (d) Distribution of reconstructed hit energies in fC.

The DQM processing unpacks the CSC raw data information, forms digitized pulse information (digis) and 2D hits (rechits) and muon chamber track segments (using the SK algorithm). Histograms are then filled for these quantities.

The CSC muon chambers provide signals from anode wires (essentially along a constant η) and from cathode strips (essentially along constant ϕ). Each chamber has six layers of wires and strips. The readout of these components is monitored by occupancy histograms for layers (see Figure 18, top right), separately for wires and strips for each chamber. The higher level rechits as well as the track segments are monitored e.g. using the rechit distribution in the $r - \phi$ plane (see Figure 18, top left). The chambers are arranged in rings around the beam pipe (in the $r - \phi$ plane). There are four separate layers of rings along the beam pipe, referred to as Stations. Stations 1 (closest to the interaction point) through 3 (away from the interaction point) were included in MTCC-I. There are three rings of chambers in Station 1 and two rings in Station 2 and 3. The occupancy of a complete ring is monitored using the histogram shown in Figure 16a.

The data for one entire run is split into several files. The analysis job is run on each file separately. After all single jobs have finished, the resulting ROOT files containing the output histograms are subsequently merged into a single ROOT file per run. This procedure is described in more detail in Section 3.1.

In an additional step, summary plots are produced based on the merged output histograms.

The final set of histograms are made globally available using the WBM tools, for example run 2643:

`http://nippon.fnal.gov/cmsdb/servlet/DQMBrowser?RUN=2643`.

4.5 Basic CSC Quasi-Online DQM Instructions

Run the CSC quasi-online DQM script to produce histograms for run NNNN

The DQM jobs are started automatically every 15 minutes by the cron job script described in Section 3.1.

In the event that one may need to run it manually, several steps should be followed:

1. As user `cmsroc`, log onto the `cmsuaf.fnal.gov`
2. cd /uscmst1b_scratch/lpc1/cmsroc/MTCC/MTCC/muon/
3. /uscmst1b_scratch/lpc1/cmsroc/MTCC/MTCC/muon/process_muon_fbf.py NNNN y
   where NNNN is the run number, e.g. 2526, and the "y" is "yes to submit" to submit the job to the Condor batch queue. When doing a test, use "n" for "no to submit".

One can check on whether the jobs are running using "condor_q -dag" on the same machine. One can run the script even if the run files have not yet fully arrived at FNAL. The same script can be rerun again, and only the new or remaining runs will be processed and added to the output.

Checking histograms

After about five minutes, depending on the load on the Condor farm, the first DQM output should be available from the RunSummary web page for run NNNN. Once the Muon CSC DQM job has finished, one should find the file:

+Final_muon_0000NNNN.root

Use a left mouse click, and then scroll all the way down. In case the merger has not worked, one should try to run the merger manually, and replace NNNN with the four digit run number:

`/uscmst1b_scratch/lpc1/cmsroc/MTCC/DQM/muon/scripts/Muon_0000NNNN_merger.csh`

Minimal check

There are four essential histograms to monitor.

Station occupancy: globESRMult
There should be entries for the three Stations (1, 2 and 3) that are in the MTCC-I. They show up as entries around the values of 2, 6 and 10. Figure 16 shows an example plot for run 2526

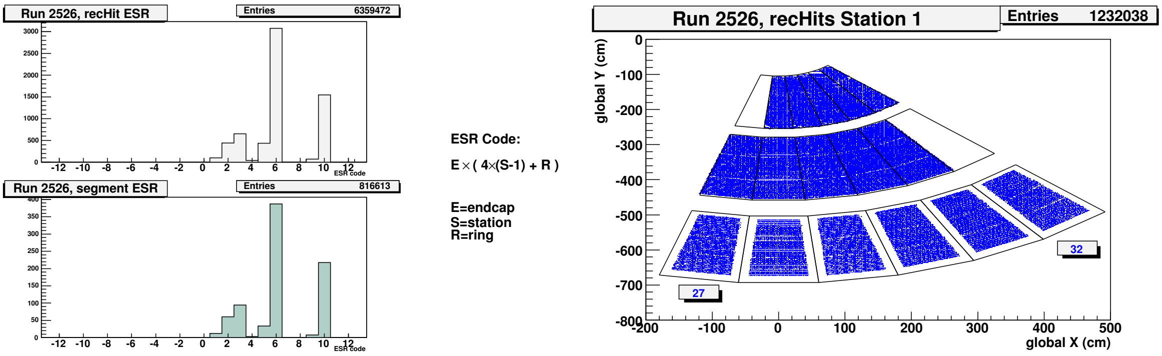


Figure 16: Run 2526: a) Global Station hit occupancy and b) $r - \phi$ distribution of the reconstructed hits of the chambers in Station 1.

Reconstructed hits on all chambers in Station 1: globReHitSt1
The chambers in Station 1 should have an even occupancy. If the chambers are empty, the low voltage may be off. One should check in the logbook to see if this was intended. Figure 17a shows an example plot for run 2526.

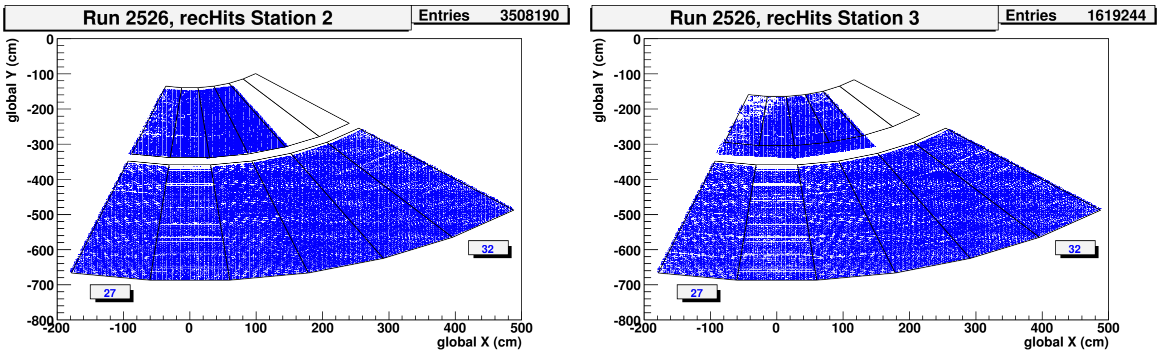


Figure 17: Run 2526, $r - \phi$ distribution of the reconstructed hits of chambers in a) Station 2 and b) Station 3.

Reconstructed hits on all chambers in Station 2: globReHitSt2
Same as for Station 1.

Reconstructed hits on all chambers in Station 3: globReHitSt3
Same as for Station 1.

Extended check

All of the layers in the chambers need to be checked to see that they are working by scanning through the plots from recHitPlotsME11Chamber27 to recHitPlotsME32Chamber31. Figure 18 shows an example plot for run 2526. The top right corner shows the entries per layer where all six bins should be filled.

4.6 Event Display

An important component in monitoring the quality of data in any experiment is the “event display”. This software provides a visual interpretation of the data and detector on an event by event basis. In CMS this is done through the “Interactive Graphics for User Analysis” (IGUANA) [7, 8, 9, 10] visualization toolkit. This system interfaces with the CMSSW software framework to reconstruct events, and provides a graphical user interface (GUI) to examine the events in both two and three dimensions.

Examples of the IGUANA event display are shown in Figures 19 and 20. Figure 19 shows a three dimensional view of the CMS detector with detector hits superimposed. This display may be rotated along any axis, magnified, clipped, sliced and viewed in multiple ways. Lightly moving the mouse and releasing the mouse button would set the detector slowly spinning around an axis. IGUANA is often set to this configuration for display purposes during off hour at the FNAL ROC. Figure 20 shows the two dimensional “R-phi” view of the detector with the hits from a muon track clearly seen in the DT detectors. This event was taken during 4.1 Tesla run, and the curve due to the magnetic field is clearly seen.

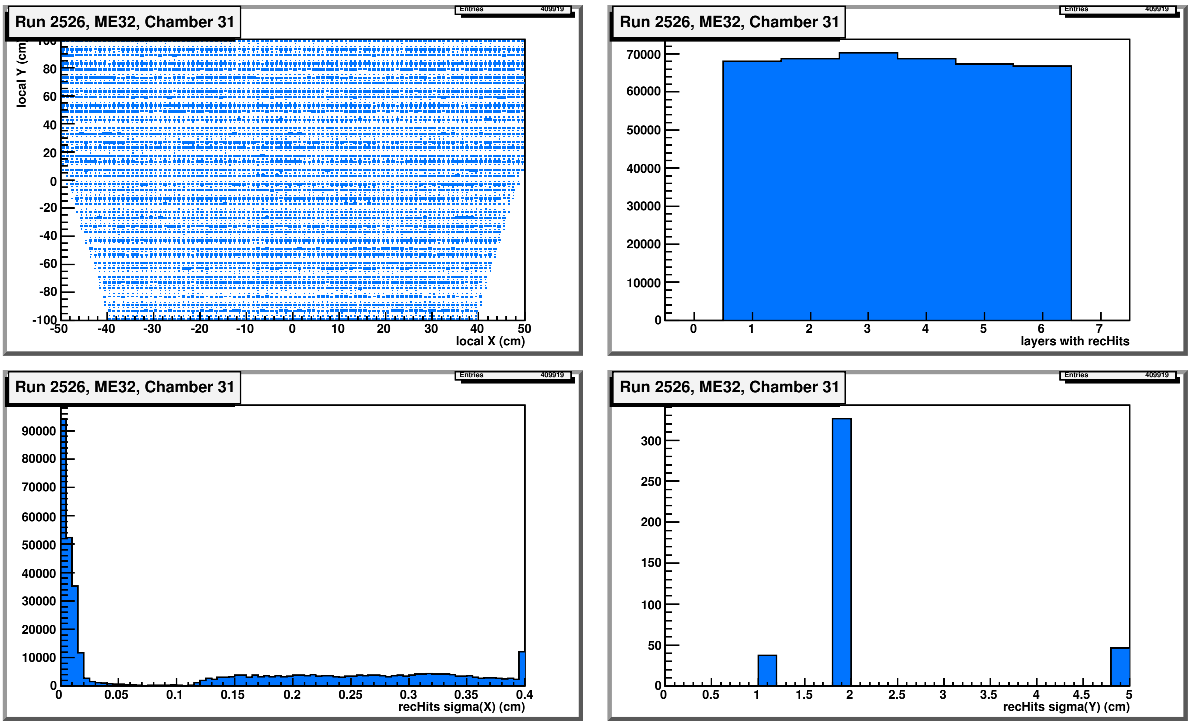


Figure 18: Run 2526, Station 3, Ring 2, Chamber 31: Shown are a) distributions of rechits in the $r - \phi$ plane, b) layer hit occupancy, c) local error on the rechit position in cm in the local x coordinate (roughly global ϕ) and d) local error on the rechit position in cm in the local y coordinate (roughly global r).

4.6.1 The runeventdisplay script

While code development proceeded and subdetectors came online, using IGUANA to display MTCC-I data was challenging. A simple Perl [34] script, called *runeventdisplay*, was written to automate the setup and execution of IGUANA.

The Perl script is available in the LPC CVS archive. Given a run number as an argument, the script would set up the appropriate environment variables and prepare the CMSSW parameter set necessary for event reconstruction. This was done by reading in a prototype parameter set file containing keys which were then replaced with the list of files to be processed. All available ROOT filenames for a desired run would be placed in the generated parameter set file, and IGUANA would be instructed to run over all events contained therein.

At the FNAL ROC, IGUANA would usually be used to display the latest runs transferred to Fermilab. An IGUANA session typically proceeded as follows. At startup, a new window would appear, and a series of libraries would be loaded to construct the detector geometry and prepare for event reconstruction. This process would take approximately two minutes. The user would then be presented with a blank display window, from which the user must manually click within the "Event" pull-down menu to load an event. Initial event reconstruction typically would take another minute or two, after which a view of the detector and reconstructed event hits would become visible. At this point the user would be able to manually customize the display. Detector and reconstructed hit elements were available to be turned on and off via the "twig menu", or check boxes, in the left panel of the main IGUANA window. During MTCC-I, the check boxes would need to be re-checked or unchecked each time IGUANA is started, and no mechanism was available to save the current state. An environment variable: `IGUANA_CONFIG`, when set to "`AUTO_START`", pre-activated a fixed set of the check boxes via hooks imbedded in the actual CMSSW code. This coupled with instability in the reconstruction code made IGUANA operation dominated by manually manipulating check boxes and waiting for libraries to be loaded.

4.6.2 Feedback and Requests for MTCC-II

Instability and need for manual reconfiguration are probably the features in most need of improvement in future versions of IGUANA. For example, in order to provide useful debugging feedback to developers, it is important to verify whether problems are reproducible in some way, and as clearly as possible understand the steps taken before the problem occurred. This is very difficult when configuration relies on a long series of manual steps. Considerable time was spent preparing IGUANA for startup that could better be spent using the displays to evaluate

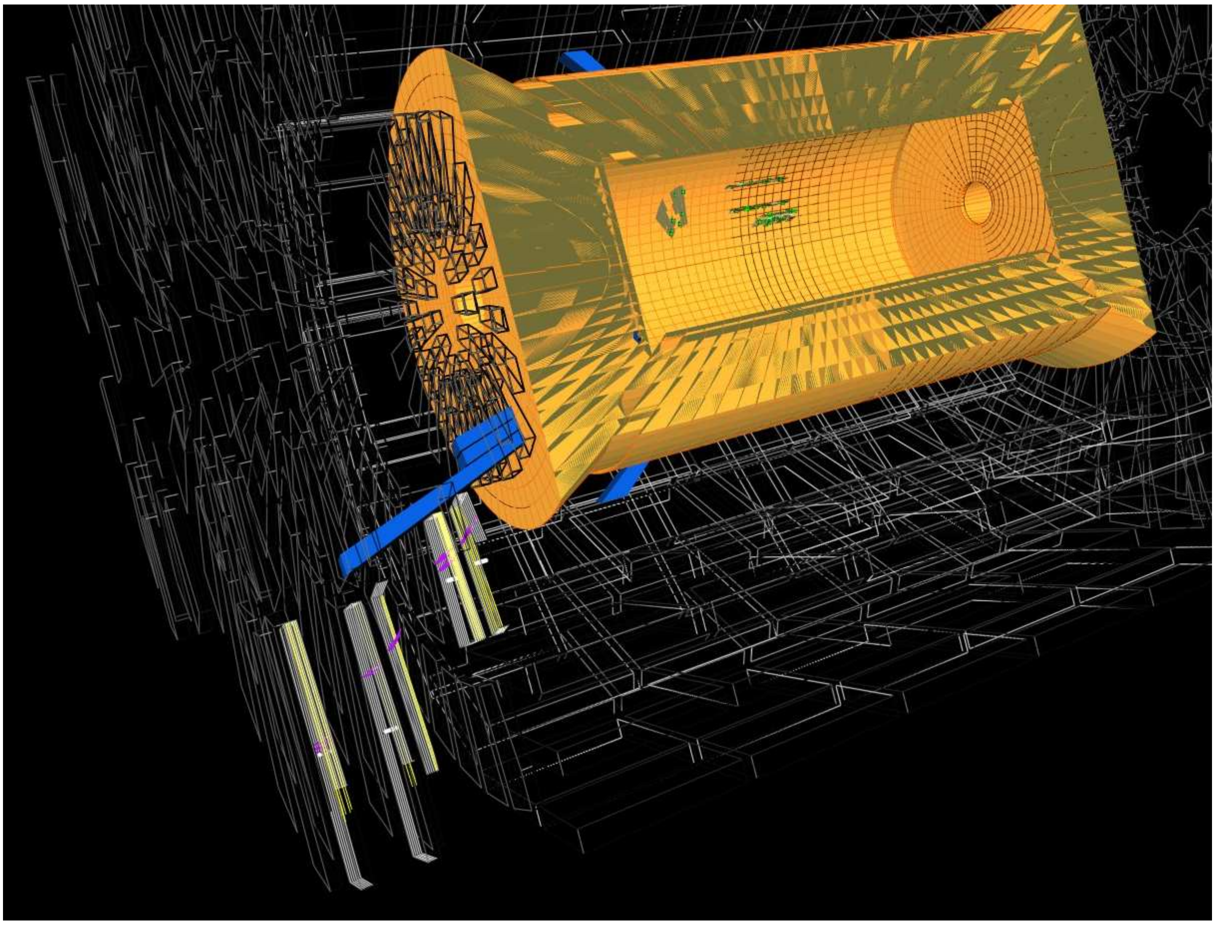


Figure 19: IGUANA 3D display showing a clip view of the detector for an event from run 2631. Hits in the tracker are visible in green, and calorimeter towers are shown as blue bars with length proportional to the energy deposited. The CSC hits are highlighted with the muon track hits in purple. The ECAL was only available at the very end of MTCC-I, and is not shown for this run.

data quality.

During MTCC-I, the data quality monitoring was limited to the level of determining whether data was present in a given subdetector or not. Pedestals and calibrations were taken from more or less random files made available sporadically by people working locally at CERN, and it was never clear whether artifacts seen in the event display were real or due to incorrect pedestal data. This will likely be improved with online and offline database access. There also was no simple mechanism for data selection criteria such as triggers fired, numbers of tracks, reconstructed energies, etc. to assist in subdetector debugging.

Despite the clear need for further development, IGUANA proved to be a useful tool for monitoring the detector and data during MTCC-I. Further, being able to visually see the hits in near real time from cosmic rays passing through each of the subdetectors, projected onto the big screen in the FNAL ROC helped add to the excitement of the MTCC. This reminded everyone that in the near future we will be able to see real physics events displayed from the CMS detector.

5 Communications

Communication from the FNAL ROC with the CERN ROC (primarily Martijn Mulders), the T0/T1 facility people, CERN MTCC operations, and the Point-5 data-taking crew was done in multiple ways:

- Scheduled meetings.
  - Daily at 9 am from the FNAL ROC. We were often connected by phone or VRVS with Martijn Mulders who was conducting online shifts at CERN from his Meyrin office.

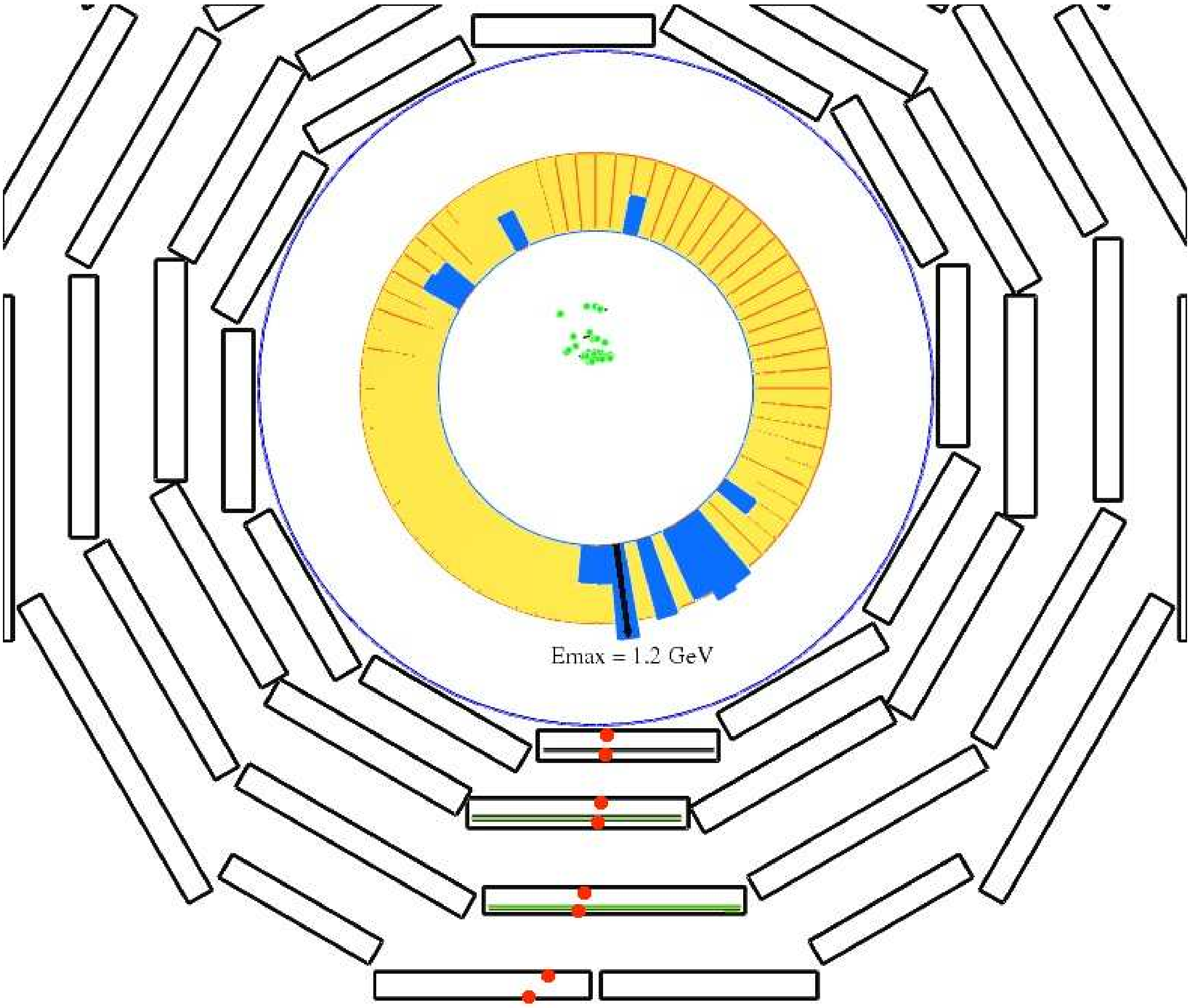


Figure 20: An example of a muon track with hits highlighted in red curving in the 4.1 T magnetic field at the end of MTCC-I.

   – Weekly meetings hosted at CERN via VRVS. The CPT MTCC, EVF/DQM and Run Meetings are scheduled on Fridays. Reports from the FNAL ROC were given in those meetings.

- We made entries in the CMS ELog at CERN, as well as the as the FNAL ROC ELog.
- TWiki [35] pages were setup to coordinate and document the MTCC quasi-online data monitoring at both CERN and the FNAL ROC [36].
- Web pages were used extensively for book-keeping and for displaying information such as the run state.
- A web camera was available from the FNAL ROC. VRVS was used continuously during the last weekend of data taking to display the activity from inside the Green Barracks.
- Frequent e-mail exchanges and occasional international phone calls.

6 Summary and Future Plans

We have successfully established a foundation for MTCC quasi-online data monitoring at the FNAL ROC which will be useful in the future for both commissioning and real beam data taking. The raw data was transferred to FNAL using the T0/T1 facility.

We ran the DQM programs for CSC, Trigger, Tracker and HCAL systematically and automatically using scripts we created. We also ran the IGUANA event display as soon as events were transferred and available to us.

We have made all our DQM results readily available by linking them from the RunSummary web page, which can be easily viewed from anywhere in the world with just a web browser. We have also demonstrated the same scheme is easily applied to the real time online monitoring viewing by linking the High-Level Trigger Local Trigger Control monitoring results through the RunSummary web page.

Coordinating with our CERN CPT MTCC colleagues, we have successfully taken a substantial part of quasi-online monitoring shifts from the FNAL ROC over the major MTCC-I 3.8 - 4T runs in the last week of August.

We have experienced and identified various difficulties and issues to be improved upon for MTCC-II and beyond.

The FNAL ROC group will continue to develop and support tools useful for taking high quality data (e.g. WBM) by coordination and agreement with the groups at CERN. The tools we are developing are useful for the entire CMS collaboration as demonstrated in MTCC-I.

Since the FNAL ROC is near the end of the data delivery chain, we would like to use our experience to make the data quality in each run (and data sets in the future) more transparent to users.

In the short term, we plan to participate in the MTCC-II activities. We would like to continue to refine what we have established in MTCC-I, and we hope to do more real-time data monitoring. In early 2007, we expect to assist with the Tracker Integration test.

In the longer term, we will continue to develop and test tools wich are needed for monitoring, and to participate and to contribute in the CMS physics data taking.

7 Acknowledgement

We would like to acknowledge the T0/T1, Event Filter, DQM, CMSSW, DAQ/Trigger, CPT-MTCC and MTCC Operations groups for their excellent work due to which we were able to perform the quasi-online data quality monitoring at the FNAL ROC. In addition, we would like to thank the following individuals with their assistance in communication, computing and software, data transfer and storage, and general expertise, all of which was very important to our success: Jon Bakken, Austin Ball, Kurt Biery, Eric Cano, Harry Cheung, Tim Christiansen, Pawel de Barbaro, Albert de Roeck, Ian Fisk, Pavel Goglov, Eric James, Christos Leonidopoulos, Dimitrije Maletic, Emilio Meschi, Martijn Mulders, Filip Moortgat, Steve Murray, Alexander Oh, Ianna Osborne, Fedor Ratnikov, Frederic Ronga, Michael Schmitt, Ilaria Segoni, Michal Szleper, Dimitris Tsirigkas, Lassi Tuura, Tony Wildish and Peter Wittich.

A Lessons Learned From MTCC-I

In a short amount of time we managed to set up an automatic quasi-online processing of MTCC data files as soon as they were transferred from SX5 to CASTOR (at Meyrin) and subsequently to dCache (at Fermilab). At the Meyrin site a simple analyzer ran in batch jobs on lxplus verifying data integrity and checking basic event properties such as data size and reported bunch crossing numbers for all of the front-end detector (FED) raw data in the events. At the FNAL ROC, four DQM analyzers and the event display ran systematically.

All of the results and histograms from the analyzed runs were made visible through the RunSummary web page:

`http://cmsdaq.cern.ch/cmsmon/cmsdb/servlet/RunSummary.`

Activities by the CPT-MTCC team at CERN and the FNAL ROC were closely coordinated from the beginning. During the last few days of MTCC-I, the CPT shifts were organized around the clock in close collaboration between CERN and the FNAL ROC, and documented from a dedicated Twiki page:

`https://twiki.cern.ch/twiki/bin/view/CMS/CPTMTCCDataMonitoringShifts.`

The simple analyzer at Meyrin processed more than 25 million events without problems. An overview of processed files and corresponding results were available at all times from:

`http://cmsdaq.cern.ch/cmsmon/cmsmtcc/runs.html,` with a minimum delay of fifteen minutes when data transfer conditions were ideal.

On several occasions integrity and/or quality problems were spotted quickly and communicated to the relevant sub-detector experts who could fix the issue, thus contributing to the MTCC data quality.

On other occasions fast processing was hampered by slow data transfer out of SX5, or difficulties accessing the data in CASTOR. In these cases we were able to provide early warnings of the transfer problems and log files with relevant details. Our bug reports of CASTOR access difficulties also led to the discovery of a misconfiguration of CASTOR access from the SX5 side, which was fixed within a few days leading to a marked improvement in data transfer and access. Remaining CASTOR limitations at (CERN-wide) peak times are being addressed in common meetings between computing and (IT) CASTOR people. It is expected that they will be solved in the next few months, but not in time for MTCC-II.

At the FNAL ROC, we successfully exercised and established quasi-online MTCC data monitoring which involved:

- Data Transfer (the T0/T1 facility) [SX5 → CASTOR(PhEDEx) → dCache]
- Automatic and systematic running of the event display, three DQM programs (HCAL, Trigger, Tracker) and a CSC analysis module in a quasi-online mode, as files arrived at Fermilab.
- Making the histogram results of the DQM programs readily and easily available during a run and also for all runs which were taken for about 25 million events. This was made possible by providing the RunSummary web page and linking the results with a built-in ROOT histogram browser capability. Therefore, there is no need to run a special application program in order to view DQM output histograms.
- In order to accomplish the above, we needed to write a script which performed multiple functions including the conversion of all of the incoming files from the .dat format to the .root format. The status of the quasi-online processing at the FNAL ROC can be viewed at:

  `http://nippon.fnal.gov/lpc1/cmsroc/MTCC/check_mtcc/.`

- Quasi-online DQM monitor results via the RunSummary web page can be also adapted to show the results from the real-time online DQM running in SX5. During MTCC-I, we made the results of LTC monitor, which was running on the HLT farm, available from the RunSummary web page. As more and more online monitors begin running at SX5, they can be easily added in the similar way.

Aspects of MTCC-I that could have been better:

- As mentioned above, data transfer was absolutely critical. A delay of 10 minutes is fine, but a delay of an hour or more would degrade the usefulness of quasi-online monitoring. It happened several times that the delay was (much) more than that.
- During the last weekend of MTCC-I, the data transfer logic out of SX5 was changed from FIFO to LIFO (the newest files are transferred first). This improved the shortest delay times, but made it very difficult for offline analyzers to get an overview of what data is still missing. It would be useful to have a tool which would keep track of what files are still waiting for transfer in the online disk buffer at SX5.
- On one occasion transfer was interrupted for half a day because a change of configuration on the SX5 side was not communicated to the computing people at the Meyrin side. Improved communication is desirable.
- The additional data transfer latency from CERN to Fermilab depends on the performance of CASTOR, PhEDEx and dCache. At good times the overall latency was less than thirty minutes from the time data was taken. But a lot of files arrived much later (a day or more). One issue is that currently the transfer mechanism between Tier 0 and Tier 1 does not allow the assignment of priorities. This could cause a conflict between MTCC-II and Computing, Software and Analysis (CSA) [37] data transfer.
- Due to a known problem the transfer rate to dCache was limited to approximately 1GB/min, half of the peak rate of the DAQ at SX5. This problem was fixed after MTCC-I.
- Stability of the data transfer remains a concern for MTCC-II.

- When a new ROOT version was introduced in offline CMSSW version 0_8_0_pre4, it caused a backward incompatibility with the 0_7_0 data which was only fixed in CMSSW version 0_9_0_pre2 and higher (even though a converter was available as patch a few days later). This was very inconvenient for analyzers and experts who needed the latest reconstruction because they had to revert to 0_8_0_pre3 and add a large number of patches by hand.
- The switch from the 0_7_0 to the 0_9_0 software release online was announced only several hours after the fact. It took quasi-online processing and data transfer over twelve hours to recover from the change. The new 0_9_0 .dat format was not originally meant to be the final format in which data was to be stored in CASTOR. It still is less convenient to use in the offline analysis.
- For 'CPT-shifters' at the Meyrin site, the absence of a nice place to sit (like the FNAL ROC) was considered a disadvantage, thus reducing the effectiveness of the shifts. Although in principle the quasi-online monitoring could be performed from any workstation, one screen is not enough to see all of the relevant information at a glance: CMS Page 1, ELog, Runs.html overview of processed runs, an lxplus window, the RunSummary web page, the CPT-Shifters Twiki page, etc. At the third floor in building 40, such an area with four screens is being installed in time for MTCC-II.
- In general the CMSSW software for unpacking, DQM, reconstruction, and event display was found to be too difficult to use 'out-of-the-box'. A lot of expert help was needed to get subsystems to work. Each had its own unique issues, but one of the main common issues had to do with getting the correct external files such as calibration files, cable mapping files, etc. which may vary run by run.
- Easy access and links to the correct database (external files) from the analysis programs is needed to do online or offline monitoring correctly.

There were many positive things to take away from MTCC-I:

- It was very exciting to be part of the 'team'. Communication between the online detector experts, T0/T1 computing groups, the CPT-MTCC team at CERN and the FNAL ROC went very well.
- The VRVS connection from the Green Barracks during data taking was a nice touch.
- Many useful lessons were learned, and midway through MTCC-I visible improvements were implemented in response to the problems as they were encountered. This was very encouraging.
- A large set of good quality MTCC-I data was recorded, and the data integrity was checked quasi-online. In the coming months this data will be extremely useful to test, debug and commission our new reconstruction software algorithms and to check and improve the Monte Carlo simulation. The data has already been useful for testing more sophisticated algorithms we want to include in MTCC-II.

References

[1] The draft version (version 2) is available from the run meeting agenda server.
http://indico.cern.ch/getFile.py/access?contribId=3&resId=1&materialId=7&confId=4249.

[2] Collaboration of US scientists participating in the CMS experiment.
http://www.uscms.org.

[3] The LHC@FNAL center on the first floor of the Wilson Hall at Fermilab will combine the remote operations of the FNAL ROC and LHC accelerator groups.
http://cd-amr.fnal.gov/remop/remop.html.

[4] W. Badgett, F. Cavallari, R. Egeland, F. Glege, K. Maeshima, S. Murray, Z. Wan, "Proposal for Web-Based Monitoring and Database Browsing", CMS note number: CMS IN-2006/044.

[5] http://cmsdaq.cern.ch/cmsmon/cmsdb/servlet/RunSummary.

[6] M. Sanders, "HCAL Calibration and Test-Beam Data at Fermilab", CMS note number: CMS IN-2006/XXX.

[7] CMS Physics Technical Design Report, Volume 1, Detector Performance and Software, CERN/LHCC 2006-001, CMS TDR 8.1, 2 February 2006.

[8] http://iguana.web.cern.ch/iguana/.

[9] G. Alverson, G. Eulisse, S. Muzaffar, I. Osborne, L. A. Tuura and L. Taylor, "IGUANA Architecture, Framework and Toolkit for Interactive Graphics," eConf **C0303241**, MOLT008 (2003), [arXiv:cs.se/0306042].

[10] G. Alverson, I. Osborne, L. Taylor, and L. A. Tuura, "A Coherent and Non-Invasive Open Analysis Architecture and Framework with Applications in CMS," *Proceedings of Computing in High Energy*, (CHEP 2001) Beijing, 3-7 Sept, 2001.

[11] https://cmsdoc.cern.ch/cms/cpt/Software/html/General.

[12] http://www.dcache.org.
See also http://www.uscms.org/SoftwareComputing/UserComputing/MassStorage.html.

[13] http://www-stken.fnal.gov/enstore/.

[14] http://cms-project-phedex.web.cern.ch/cms-project-phedex.

[15] http://sdm.lbl.gov/srm-wg/index.html.

[16] http://castor.web.cern.ch/castor/.

[17] https://srm.fnal.gov/twiki/bin/view/SrmProject/WebHome.

[18] https://twiki.cern.ch/twiki/bin/view/CMS/FilterUnitOutputHandling.

[19] https://twiki.cern.ch/twiki/bin/view/CMS/DBS-TDR.

[20] http://cmsdaq.cern.ch/cmsmon/cmsmtcc/runs.html.

[21] http://www.python.org/.

[22] http://www.cs.wisc.edu/condor/.
See also http://cmssrv02.fnal.gov/condor/state.html.

[23] http://en.wikipedia.org/wiki/Cron.

[24] http://root.cern.ch/.

[25] http://nippon.fnal.gov:8081/cmsroc/.

[26] http://cdcvs.fnal.gov/cgi-bin/public-cvs/cvsweb-public.cgi/MTCC/?cvsroot=lpc.

[27] http://en.wikipedia.org/wiki/Datagram_Congestion_Control_Protocol.

[28] http://www.uscms.org/SoftwareComputing/UserComputing/UserComputing.html.

[29] http://tomcat.apache.org/.

[30] http://jakarta.apache.org/.

[31] Physics and Data Quality Monitoring Infrastructure.
http://cmsevf.web.cern.ch/cmsevf/DQM_doc/DQM_instructions.html.

[32] C. Leonidopoulos, E. Meschi, I. Segoni, G. Eulisse, D. Tsirigkas, "Physics and Data Quality Monitoring at CMS," ICHEP06 (2006).
http://indico.cern.ch/materialDisplay.py?contribId=244&sessionId=2&materialId=paper&confId=048.

[33] http://lotus.phys.northwestern.edu/twiddleschmittm/CMS/LPC-muon/most_recent_CSC_example.html.

[34] http://www.perl.com.

[35] https://twiki.cern.ch/twiki/bin/view/Main/WebHome.

[36] https://twiki.cern.ch/twiki/bin/view/CMS/CPTMTCCDataMonitoringShifts.

[37] https://uimon.cern.ch/twiki/bin/view/CMS/CSA06.